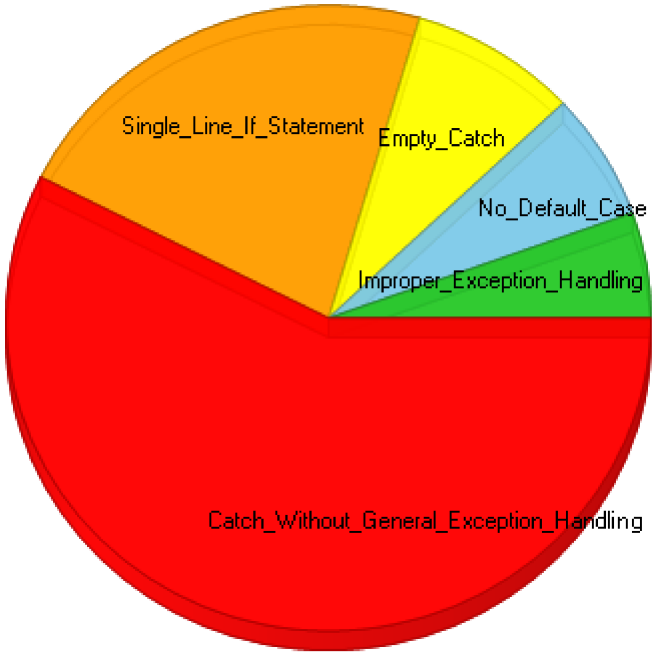
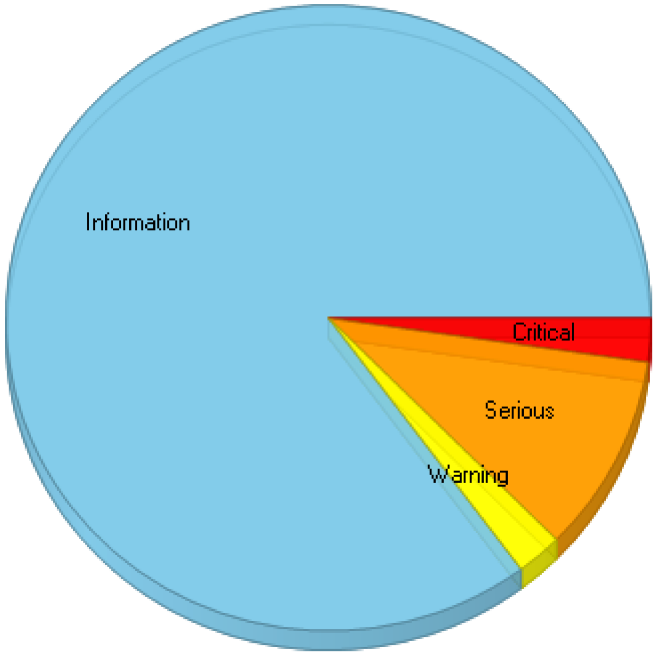
Checkmarx Report

Project Name:	**MvnForum**
Load Time:	**27/02/2008 11:55:00**
CxSuite Version:	**2.4.3**

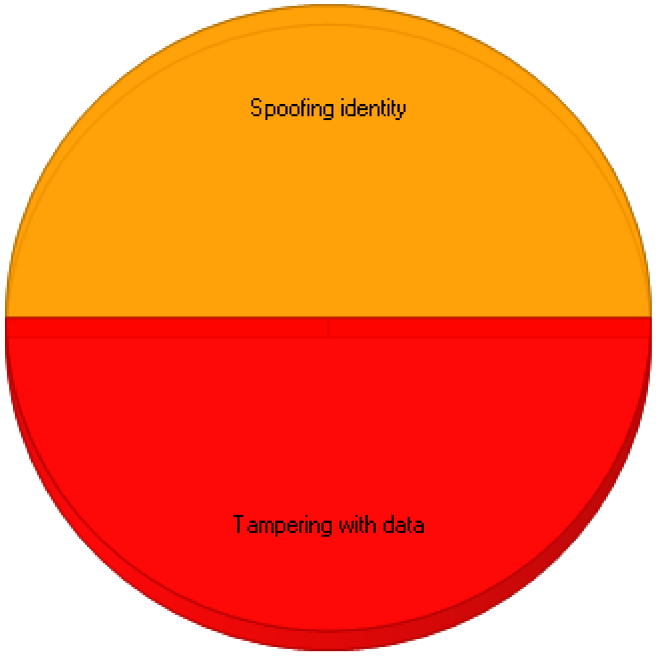
Problems by Queries



Problems by Severity



Problems by Impacts



Problems by Files

Summary

Query	Group	Problems found	Severity
Improper Exception Handling	Best Coding Practice	154	
Unclosed Objects	Best Coding Practice	39	
Catch Without General Exception Handling	Best Coding Practice	1621	
Single Line If Statement	Best Coding Practice	624	
Empty Catch	Best Coding Practice	237	
No Default Case	Best Coding Practice	186	
Erroneous String Compare	Best Coding Practice	2	
Magic Numbers	Best Coding Practice	61	
Confusing Naming	Best Coding Practice	16	
Leftover Debug Code	Best Coding Practice	13	
Missing Catch Block	Best Coding Practice	5	
Missing XML Validation	Best Coding Practice	7	
Overly Broad Catch	Best Coding Practice	110	
Overly Broad Throws	Best Coding Practice	16	
Use of System Output Stream	Best Coding Practice	81	
SQL injection	High Risk	3	
Second Order SQL Injection	High Risk	63	
Reflected XSS All Clients	High Risk	12	
Stored XSS	High Risk	6	
Log Forgery	Low Visibility	18	
Password Misuse	Low Visibility	53	
DoS by Unreleased Resources	Low Visibility	7	
Hardcoded Password	Low Visibility	8	
Verbose Error Reporting	Medium Threat	90	
Files Manipulation	Medium Threat	20	
XSRF	Medium Threat	135	
Parameter Tampering	Medium Threat	132	
Blind SQL Injections	Medium Threat	3	
Singleton HTTPServlet	Medium Threat	2	

Metrics

Metric Name	Value
Number of types	915
Number of Classes	784
Number of Interfaces	131
Avg methods per class	10
Number of Constructors	603
Avg statements per Method	8
Number of Control Statements	13894
Number of Methods	8167
Lines of Code	201594

Top 10 Files

File name	Problems found
mvnforum-1.1-src\myvietnam\src\net\myvietnam\mvncore\util\genericparamutil.java	184
mvnforum-1.1-src\mvnforum\contrib\phpbb2mvnforum_method1\src\migrator.java	155
mvnforum-1.1-src\mvnforum\src\com\mvnforum\db\jdbc\memberdaoimpljdbc.java	111
mvnforum-1.1-src\mvnforum\contrib\phpbb2mvnforum-jaxb\src\com\mvnforum\jaxb\db\impl\membertypeimpl.java	106
mvnforum-1.1-src\mvnforum\src\com\mvnforum\auth\onlineusermanager.java	103
mvnforum-1.1-src\mvnforum\src\com\mvnforum\db\jdbc\threaddaoimpljdbc.java	84
mvnforum-1.1-src\mvnforum\src\com\mvnforum\admin\memberxml.java	64
mvnforum-1.1-src\mvnforum\src\com\mvnforum\db\jdbc\postdaoimpljdbc.java	63
mvnforum-1.1-src\mvnforum\src\com\mvnforum\db\jdbc\forumdaoimpljdbc.java	61
mvnforum-1.1-src\mvnforum\src\com\mvnforum\user\threadwebhandler.java	59

	Often Misused: Exception Handling
CWE ID	248
Description	Failing to catch an exception thrown from a dangerous function, can potentially cause the program to crash.
Demonstrative Examples	The _alloca() function allocates memory on the stack. If an allocation request is too large for the available stack space, _alloca() throws an exception. If the exception is not caught, the program will crash, potentially enabling a denial of service attack. _alloca() has been deprecated as of Microsoft Visual Studio 2005®. It has been replaced with the more secure _alloca_s().
	EnterCriticalSection() can raise an exception, potentially causing the program to crash. Under operating systems prior to Windows 2000, the EnterCriticalSection() function can raise an exception in low memory situations. If the exception is not caught, the program will crash, potentially enabling a denial of service attack.
Node Relationships	ChildOf - API Abuse (227) Peer - Error Conditions, Return Values, Status Codes (389)
Source Taxonomies	7 Pernicious Kingdoms - Often Misused: Exception Handling
Applicable Platforms	C C++ Java .NET

Path 1:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
101. public static final void saveFile(String filename, String text) throws
IOException //utility.java
     ...
103. out.write(text);
```

[Table of Contents] [Description]

Path 2:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
115. public static final void saveFile(String filename, StringBuffer text) throws
IOException //utility.java
```

```
...
117. out.write(text.toString());
```

[Table of Contents] [Description]

Path 3:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
    ...
    107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
    where user_id =" + id);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
    ...
    110. memname = mrs.getString("username");
```

[Table of Contents] [Description]

Path 5:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
    125. ResultSet resultSet = statement.executeQuery("SELECT user_timezone from
    phpbb_users where user_id =" + id);
```

[Table of Contents] [Description]

Path 6:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
    129. timeZone = resultSet.getInt("user_timezone");
```

[Table of Contents] [Description]

Path 7:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
      |143. ResultSet RS = Stmt.executeQuery("SELECT * FROM phpbb_users");
```

[Table of Contents] [Description]

Path 8:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
      |150. String MemberID = RS.getString("user_id");
```

[Table of Contents] [Description]

Path 9:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
      |151. String MemberName = RS.getString("username");
```

[Table of Contents] [Description]

Path 10:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
      |153. int timeZone = RS.getInt("user_timezone");
```

[Table of Contents] [Description]

Path 11:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
```

```
163. String MemberPassword = Utils.wrapit(MD5.getBase64FromMD5(RS.getString
("user_password")));
```

[Table of Contents] [Description]

Path 12:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
164. String MemberFirstEmail = Utils.wrapit(RS.getString("user_email"));
```

[Table of Contents] [Description]

Path 13:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
167. String MemberEmailVisible = RS.getString("user_viewemail");
```

[Table of Contents] [Description]

Path 14:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
173. String MemberPostCount = RS.getString("user_posts");
```

[Table of Contents] [Description]

Path 15:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
177. String MemberCreationDate = Utils.HauDateTimeFromS(RS.getLong
("user_regdate"), timeZone);
```

[Table of Contents] [Description]

Path 16:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
184. String MemberLastLogon = Utils.HauDateTimeFromS(RS.getLong
("user_lastvisit"), timeZone);
```

[Table of Contents] [Description]

Path 17:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
203. String MemberSignature = Utils.wrapit(RS.getString("user_sig"));
```

[Table of Contents] [Description]

Path 18:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
205. String MemberAvatar = Utils.wrapit(RS.getString("user_avatar"));
```

[Table of Contents] [Description]

Path 19:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
207. String MemberLanguage = Utils.wrapit(RS.getString("user_lang"));
```

[Table of Contents] [Description]

Path 20:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
216. String MemberBirthday = Utils.DateFromS(RS.getLong("user_regdate"));
```

[Table of Contents] [Description]

Path 21:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
219. String MemberAddress = Utils.wrapit(RS.getString("user_from"));
```

[Table of Contents] [Description]

Path 22:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
226. String MemberCareer = Utils.wrapit(RS.getString("user_occ"));
```

[Table of Contents] [Description]

Path 23:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
228. String MemberHomepage = Utils.wrapit(RS.getString("user_website"));
```

[Table of Contents] [Description]

Path 24:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
229. String MemberYahoo = Utils.wrapit(RS.getString("user_yim"));
```

[Table of Contents] [Description]

Path 25:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
```

```
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
230. String MemberAol = Utils.wrapit(RS.getString("user_aim"));
```

[Table of Contents] [Description]

Path 26:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
232. String MemberMsn = Utils.wrapit(RS.getString("user_msnm"));
```

[Table of Contents] [Description]

Path 27:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
236. String MemberCoolLink1 = Utils.wrapit(RS.getString("user_interests"));
```

[Table of Contents] [Description]

Path 28:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 29:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
    ...
230. String MemberAol = Utils.wrapit(RS.getString("user_aim"));
```

[Table of Contents] [Description]

Path 30:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      |...
      |279. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 31:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      |...
      |296. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_categories");
```

[Table of Contents] [Description]

Path 32:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      |...
      |302. String CategoryID = RS.getString("cat_id");
```

[Table of Contents] [Description]

Path 33:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      |...
      |304. String CategoryName = Utils.wrapit(RS.getString("cat_title"));
```

[Table of Contents] [Description]

Path 34:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      |...
```

```
314. String CategoryOrder = RS.getString("cat_order");
```

[Table of Contents] [Description]

Path 35:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
...
344. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 36:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
360. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_forums");
```

[Table of Contents] [Description]

Path 37:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
368. String ForumID = RS.getString("forum_id");
```

[Table of Contents] [Description]

Path 38:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
369. String CategoryID = RS.getString("cat_id");
```

[Table of Contents] [Description]

Path 39:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
371. String last_post_id = RS.getString("forum_last_post_id");
```

[Table of Contents] [Description]

Path 40:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
phpbb_posts where post_id=" + last_post_id);
```

[Table of Contents] [Description]

Path 41:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
376. String poster_id = lastpostrs.getString("poster_id");
```

[Table of Contents] [Description]

Path 42:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
382. ResultSet userstmtrs = userstmt.executeQuery("SELECT username from
phpbb_users where user_id=" +poster_id);
```

[Table of Contents] [Description]

Path 43:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
385. String LastPostMemberName = Utils.wrapit(userstmtrs.getString
```

```
("username"));
```

[Table of Contents] [Description]

Path 44:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
390. String ForumName = Utils.wrapit(RS.getString("forum_name"));
```

[Table of Contents] [Description]

Path 45:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
391. String ForumDesc = Utils.wrapit(RS.getString("forum_desc"));
```

[Table of Contents] [Description]

Path 46:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
400. String ForumOrder = RS.getString("forum_order");
```

[Table of Contents] [Description]

Path 47:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
404. String ForumStatus = RS.getString("forum_status");
```

[Table of Contents] [Description]

Path 48:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
407. String ForumThreadCount = RS.getString("forum_topics");
```

[Table of Contents] [Description]

Path 49:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
408. String ForumPostCount = RS.getString("forum_posts");
```

[Table of Contents] [Description]

Path 50:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 51:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
483. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_posts");
```

[Table of Contents] [Description]

Path 52:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
489. String PostID = RS.getString("post_id");
```

[Table of Contents] [Description]

Path 53:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
490. String ForumID = RS.getString("forum_id");
```

[Table of Contents] [Description]

Path 54:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
491. String ThreadID = RS.getString("topic_id");
```

[Table of Contents] [Description]

Path 55:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
492. String MemberID = RS.getString("poster_id");
```

[Table of Contents] [Description]

Path 56:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
495. ResultSet ppidrs = ppidstmt.executeQuery("SELECT post_id from phpbb_posts
where topic_id=" + ThreadID + " ORDER by post_id");
```

[Table of Contents] [Description]

Path 57:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
```

```
ostream) throws SQLException //migrator.java
    ...
500. String ppidtmp = ppidrs.getString("post_id");
```

[Table of Contents] [Description]

Path 58:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
513. ResultSet userstmtrs = userstmt.executeQuery("SELECT
username,user_timezone from phpbb_users where user_id=" + MemberID);
```

[Table of Contents] [Description]

Path 59:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
515. String MemberName = Utils.wrapit(userstmtrs.getString("username"));
```

[Table of Contents] [Description]

Path 60:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
517. int timeZone = userstmtrs.getInt("user_timezone");
```

[Table of Contents] [Description]

Path 61:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 62:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
528. String PostTopic = Utils.wrapit(posttextrs.getString("post_subject"));
```

[Table of Contents] [Description]

Path 63:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
532. String PostBody = posttextrs.getString("post_text");
```

[Table of Contents] [Description]

Path 64:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
555. String PostCreationDate = Utils.HauDateTimeFromS(RS.getLong("post_time"),
timeZone);
```

[Table of Contents] [Description]

Path 65:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
561. String PostCreationIP = Utils.wrapit(Utils.HexIPtoString(RS.getString
("poster_ip")));
```

[Table of Contents] [Description]

Path 66:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
563. String PostEditCount = RS.getString("post_edit_count");
```

[Table of Contents] [Description]

Path 67:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 68:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
```

[Table of Contents] [Description]

Path 69:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
625. String ThreadID = RS.getString("topic_id");
```

[Table of Contents] [Description]

Path 70:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
627. String ForumID = RS.getString("forum_id");
```

[Table of Contents] [Description]

Path 71:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        628. String MemberName = Utils.wrapit(usernamefromid(php, RS.getString
        ("topic_poster")));
```

[Table of Contents] [Description]

Path 72:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
        where post_id = " + RS.getString("topic_last_post_id"));
```

[Table of Contents] [Description]

Path 73:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
        where post_id = " + RS.getString("topic_last_post_id"));
```

[Table of Contents] [Description]

Path 74:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        635. LastPostMemberName = usernamefromid(php, lrs.getString("poster_id"));
```

[Table of Contents] [Description]

Path 75:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
641. String ThreadTopic = Utils.wrapit(RS.getString("topic_title"));
```

[Table of Contents] [Description]

Path 76:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
644. String first_post_id = RS.getString("topic_first_post_id");
```

[Table of Contents] [Description]

Path 77:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
646. lrs = lstmt.executeQuery("select post_text from phpbb_posts_text where
post_id = " + first_post_id);
```

[Table of Contents] [Description]

Path 78:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
650. ThreadBody = lrs.getString("post_text");
```

[Table of Contents] [Description]

Path 79:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
665. String userID = RS.getString("topic_poster");
```

[Table of Contents] [Description]

Path 80:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      668. String temp1 = Utils.HauDateTimeFromS(RS.getLong("topic_time"),
      timeZone);
```

[Table of Contents] [Description]

Path 81:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      672. long myLong = RS.getLong("topic_time");
```

[Table of Contents] [Description]

Path 82:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      685. String last_post_id = RS.getString("topic_last_post_id");
```

[Table of Contents] [Description]

Path 83:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      687. lrs = lstmt.executeQuery("select post_time,poster_id from phpbb_posts
      where post_id = " + last_post_id);
```

[Table of Contents] [Description]

Path 84:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
691. String userId = lrs.getString("poster_id");
```

[Table of Contents] [Description]

Path 85:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
694. ThreadLastPostDate = Utils.DateTimeFromS(lrs.getLong("post_time"));
```

[Table of Contents] [Description]

Path 86:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
706. String ThreadStatus = RS.getString("topic_status"); // TODO: Maybe remove
status preservation
```

[Table of Contents] [Description]

Path 87:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
707. String ThreadHasPoll = RS.getString("topic_vote");
```

[Table of Contents] [Description]

Path 88:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
708. String ThreadViewCount = RS.getString("topic_views");
```

[Table of Contents] [Description]

Path 89:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
709. String ThreadReplyCount = RS.getString("topic_replies");
```

[Table of Contents] [Description]

Path 90:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 91:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
      ...
321. Document document = reader.read(new File(strPathName + "mvnforum.xml"));
```

[Table of Contents] [Description]

Path 92:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
578. public void updateConfigStepTwo(GenericRequest
request) //configurationwebhandler.java
      ...
616. Document document = reader.read(new File(strPathName + "mvnforum.xml"));
```

[Table of Contents] [Description]

Path 93:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
681. public void updateConfigStepThree(GenericRequest
request) //configurationwebhandler.java
      ...
      712. Document document = reader.read(new File(strPathName + "mvnforum.xml"));
```

[Table of Contents] [Description]

Path 94:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
767. public void updateConfigMVNCore(GenericRequest
request) //configurationwebhandler.java
      ...
      876. Document document = reader.read(new File(strPathName + "mvncore.xml"));
```

[Table of Contents] [Description]

Path 95:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
1103. public void updateUrlPattern(GenericRequest
request) //configurationwebhandler.java
      ...
      1119. Document document = reader.read(new File(xmlFilePath));
```

[Table of Contents] [Description]

Path 96:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
197. private static void addFile(ZipOutputStream out, File file, String
relativePath) //exportwebhelper.java
      ...
      211. while ((len = in.read(buf)) > 0) {
```

[Table of Contents] [Description]

Path 97:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
197. private static void addFile(ZipOutputStream out, File file, String
relativePath) //exportwebhelper.java
      ...
      212. out.write(buf, 0, len);
```

[Table of Contents] [Description]

Path 98:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
385. outputHtml.println("<html>");
```

[Table of Contents] [Description]

Path 99:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
386. outputHtml.println("<head>");
```

[Table of Contents] [Description]

Path 100:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
387. outputHtml.println(" <title>mvnForum - Import/Restore Process</title>");
```

[Table of Contents] [Description]

Path 101:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
388. outputHtml.println(" <meta name=\"copyright\" content=\"This software is
copyrighted by MyVietnam.net and released under the terms of the GNU General
Public License GPL license. All rights reserved.\">");
```

[Table of Contents] [Description]

Path 102:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
```

```
request) //importwebhandler.java
    ...
389. outputHtml.println(" <meta name=\"Keywords\" content=\"mvnForum, server
side java, online meetings, collaboration, bulletin board, board, forum,
discussion board, jsp, servlet, java, j2se, j2ee, database, jdbc, connection
pool, mysql, MyVietnam, mvnForum, framework, web app, web application, servlet
container, tomcat, jboss, weblogic, websphere\">");
```

[Table of Contents] [Description]

Path 103:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
    ...
390. outputHtml.println(" <meta name=\"Description\" content=\"mvnForum is a
powerful Jsp/Servlet forum (discussion board) - based on Java
technology.\">");
```

[Table of Contents] [Description]

Path 104:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
    ...
391. outputHtml.println(" <meta http-equiv=\"content-type\"
content=\"text/html; charset=UTF-8\">");
```

[Table of Contents] [Description]

Path 105:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
    ...
392. outputHtml.println(" <meta http-equiv=\"Expires\" content=\"-1\">");
```

[Table of Contents] [Description]

Path 106:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
    ...
```

```
393. outputHtml.println(" <link href=\""+request.getContextPath()
+"/mvnplugin/mvnforum/css/style.css\" rel=\"stylesheet\" type=\"text/css\">");
```

[Table of Contents] [Description]

Path 107:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
...
394. outputHtml.println("</head>");
```

[Table of Contents] [Description]

Path 108:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
...
395. outputHtml.println("<body leftmargin=\"0\" topmargin=\"0\">");
```

[Table of Contents] [Description]

Path 109:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
...
396. outputHtml.println("<br/>");
```

[Table of Contents] [Description]

Path 110:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
...
397. outputHtml.println("");
```

[Table of Contents] [Description]

Path 111:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
398. outputHtml.println("<table class=\"tborder\" width=\"95%\"
cellpadding=\"3\" align=\"center\">");
```

[Table of Contents] [Description]

Path 112:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
399. outputHtml.println("<tr class=\"theader\">");
```

[Table of Contents] [Description]

Path 113:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
400. outputHtml.println(" <td align=\"center\">Messages during the
process</td>");
```

[Table of Contents] [Description]

Path 114:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
382. public static void startHtml(HttpServletRequest
request) //importwebhandler.java
     ...
401. outputHtml.println("</tr>");
```

[Table of Contents] [Description]

Path 115:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
407. public static void endHtml() //importwebhandler.java
     ...
409. outputHtml.println("</table>");
```

[Table of Contents] [Description]

Path 116:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
407. public static void endHtml() //importwebhandler.java
       ...
410. outputHtml.println("");
```

[Table of Contents] [Description]

Path 117:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
407. public static void endHtml() //importwebhandler.java
       ...
411. outputHtml.println("<br/>");
```

[Table of Contents] [Description]

Path 118:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
407. public static void endHtml() //importwebhandler.java
       ...
412. outputHtml.println("</body>");
```

[Table of Contents] [Description]

Path 119:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
407. public static void endHtml() //importwebhandler.java
       ...
413. outputHtml.println("</html>");
```

[Table of Contents] [Description]

Path 120:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
       ...
498. outputHtml.println("<tr class=\"trow1\">");
```

[Table of Contents] [Description]

Path 121:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
500. outputHtml.println("<tr class=\"trow2\">");
```

[Table of Contents] [Description]

Path 122:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 123:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
503. outputHtml.println("</tr>");
```

[Table of Contents] [Description]

Path 124:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
72. public void processAdd(GenericRequest request) //membergroupwebhandler.java
      ...
111. while ((memberName = reader.readLine()) != null) {
```

[Table of Contents] [Description]

Path 125:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
257. private void outputText(String s) throws IOException //xmlwriter.java
      ...
```

```
261. outWriter.write(s);
```

[Table of Contents] [Description]

Path 126:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
273. private void outputNewLine() throws IOException //xmlwriter.java
...
277. outWriter.write(lineSeparator);
```

[Table of Contents] [Description]

Path 127:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
273. private void outputNewLine() throws IOException //xmlwriter.java
...
278. for (int i=0; i<indentLevel; i++) outWriter.write(indentString);
```

[Table of Contents] [Description]

Path 128:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
...
312. while ((n = zipInputStream.read(buffer)) > 0) {
```

[Table of Contents] [Description]

Path 129:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
...
313. outStream.write(buffer, 0, n);
```

[Table of Contents] [Description]

Path 130:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      ...
117. out.print(key);
```

[Table of Contents] [Description]

Path 131:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      ...
118. out.print("=");
```

[Table of Contents] [Description]

Path 132:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      ...
119. out.print(value);
```

[Table of Contents] [Description]

Path 133:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      ...
123. out.println();
```

[Table of Contents] [Description]

Path 134:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
180. public Document getDocument(String prefix, String
rootName) //configurationxmldocument.java
      ...
190. return reader.read(getClass().getName());
```

[Table of Contents] [Description]

Path 135:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
53. out.println("<html>");
```

[Table of Contents] [Description]

Path 136:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
54. out.println("<head>");
```

[Table of Contents] [Description]

Path 137:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
55. out.println("<title>Hello World Servlet</title>");
```

[Table of Contents] [Description]

Path 138:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
56. out.println("</head>");
```

[Table of Contents] [Description]

Path 139:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
```

```
response) //helloworldservlet.java
    ...
57. out.println("<body>");
```

[Table of Contents] [Description]

Path 140:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
    ...
58. out.println("<h1>Hello World Servlet run now.</h1>");
```

[Table of Contents] [Description]

Path 141:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
    ...
59. out.println("</body>");
```

[Table of Contents] [Description]

Path 142:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
    ...
60. out.println("</html>");
```

[Table of Contents] [Description]

Path 143:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
182. public void close() throws IOException //compressionresponsestream.java
    ...
207. output.write(buffer, 0, bufferCount);
```

[Table of Contents] [Description]

Path 144:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
331. public void writeToGZip(byte b[], int off, int len) throws
IOException //compressionresponsestream.java
      |...
      |345. gzipstream.write(b, off, len);
```

[Table of Contents] [Description]

Path 145:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
261. public static byte[] getBytes(InputStream inputStream) throws
IOException //fileutil.java
      |...
      |266. int readLength = bufferedInputStream.read(block);
```

[Table of Contents] [Description]

Path 146:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
261. public static byte[] getBytes(InputStream inputStream) throws
IOException //fileutil.java
      |...
      |268. byteArrayOutputStream.write(block, 0, readLength);
```

[Table of Contents] [Description]

Path 147:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
331. public byte readByte() //multipartstream.java
      |...
      |337. tail = input.read(buffer, head, bufSize);
```

[Table of Contents] [Description]

Path 148:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
436. public String readHeaders() //multipartstream.java
      |...
      |457. baos.write(b[0]);
```

[Table of Contents] [Description]

Path 149:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
494. public int readBodyData(OutputStream output) //multipartstream.java
       ...
507. output.write(buffer, head, pos - head);
```

[Table of Contents] [Description]

Path 150:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
494. public int readBodyData(OutputStream output) //multipartstream.java
       ...
520. output.write(buffer, head, tail - head - pad);
```

[Table of Contents] [Description]

Path 151:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
494. public int readBodyData(OutputStream output) //multipartstream.java
       ...
528. bytesRead = input.read(buffer, pad, bufSize - pad);
```

[Table of Contents] [Description]

Path 152:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
494. public int readBodyData(OutputStream output) //multipartstream.java
       ...
537. output.write(buffer, 0, pad);
```

[Table of Contents] [Description]

Path 153:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
562. public int discardBodyData() //multipartstream.java
       ...
593. bytesRead = input.read(buffer, pad, bufSize - pad);
```

[Table of Contents] [Description]

Path 154:

Query Name - **Improper_Exception_Handling**

Severity - **Information**

```
672. private void readObject(ObjectInputStream in) //diskfileitem.java
       ...
679. output.write(cachedContent);
```

[Table of Contents] [Description]

Unclosed objects

CWE ID	10031
Description	Closing objects inside a try clause, without a matching Close statement in the finally section may lead to resouce leaking in case of exception.
Likelihood of Exploit	Low
Common Consequences	In case of exception, open objects may not be closed. Make sure that the marked Close statement, which is within a try clause, has a matching Close statement in the finally clause.
Potential Mitigations	Add a matching close statement to the finally clause.
Applicable Platforms	All

Path 1:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
160. public void convert() //phpbbtomvnforum.java
       ...
189. phpbb_c.close();
```

[Table of Contents] [Description]

Path 2:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
160. public void convert() //phpbbtomvnforum.java
       ...
191. mvnforum_c.close();
```

[Table of Contents] [Description]

Path 3:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
102. public void convert() //runningversion.java
       ...
134. phpbb_c.close();
```

[Table of Contents] [Description]

Path 4:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
102. public void convert() //runningversion.java
       ...
136. mvnforum_c.close();
```

[Table of Contents] [Description]

Path 5:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
123. public static void closeResultSet(ResultSet rs) //dbutils.java
       ...
126. rs.close();
```

[Table of Contents] [Description]

Path 6:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
137. public static void closeStatement(Statement statement) //dbutils.java
       ...
140. statement.close();
```

[Table of Contents] [Description]

Path 7:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
153. public static void closeConnection(Connection connection) //dbutils.java
       ...
156. connection.close();
```

[Table of Contents] [Description]

Path 8:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
130. public final com.sun.msv.grammar.Grammar getGrammar() throws
JAXBException //grammarinfoimpl.java
      ...
146. ois.close();
```

[Table of Contents] [Description]

Path 9:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
      ...
192. phpbb_c.close();
```

[Table of Contents] [Description]

Path 10:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
      ...
195. mvnforum_c.close();
```

[Table of Contents] [Description]

Path 11:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
264. public static ArrayList getContent(String fileName) //converturl.java
      ...
278. reader.close();
```

[Table of Contents] [Description]

Path 12:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
297. public static void ExportContentToFile(ArrayList lines, String
fileName) //converturl.java
      ...
305. out.close();
```

[Table of Contents] [Description]

Path 13:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
240. private void loadResourceMap(FileInputStream ins) throws
BuildException //localizingkeyfinder.java
      ...
286. in.close();
```

[Table of Contents] [Description]

Path 14:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
302. private void translate() throws BuildException //localizingkeyfinder.java
      ...
387. in.close();
```

[Table of Contents] [Description]

Path 15:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
449. private void loadResourceMap(FileInputStream ins) throws
BuildException //translate.java
      ...
495. in.close();
```

[Table of Contents] [Description]

Path 16:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
514. private void translate() throws BuildException //translate.java
      ...
622. in.close();
```

[Table of Contents] [Description]

Path 17:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
514. private void translate() throws BuildException //translate.java
      ...
625. out.close();
```

[Table of Contents] [Description]

Path 18:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
449. private void loadResourceMap(FileInputStream ins) throws
BuildException //translatetojstl.java
      ...
      495. in.close();
```

[Table of Contents] [Description]

Path 19:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
514. private void translate() throws BuildException //translatetojstl.java
      ...
      622. in.close();
```

[Table of Contents] [Description]

Path 20:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
514. private void translate() throws BuildException //translatetojstl.java
      ...
      625. out.close();
```

[Table of Contents] [Description]

Path 21:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
200. private static void setMVNForumHome(String home) //mvnforumconfig.java
      ...
      216. fos.close();
```

[Table of Contents] [Description]

Path 22:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
      ...
      271. fos.close();
```

[Table of Contents] [Description]

Path 23:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
      ...
301. fos.close();
```

[Table of Contents] [Description]

Path 24:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
174. inputStream.close();
```

[Table of Contents] [Description]

Path 25:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
182. outputStream.close();
```

[Table of Contents] [Description]

Path 26:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
80. public static void exportXml(String filename, HttpServletRequest
request, //exportwebhelper.java
      ...
99. xmlWriter.close();
```

[Table of Contents] [Description]

Path 27:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
```

```
...
182. try { out.close(); } catch (Exception ee) {}
```

[Table of Contents] [Description]

Path 28:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
152. public Configuration getConfiguration() throws
Exception //configurationfactory.java
...
183. input.close();
```

[Table of Contents] [Description]

Path 29:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
272. synchronized void freeConnection(Connection con) //dbconnectionmanager.java
...
283. con.close();
```

[Table of Contents] [Description]

Path 30:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
376. synchronized boolean release() //dbconnectionmanager.java
...
382. con.close();
```

[Table of Contents] [Description]

Path 31:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
305. public static void closeConnection(Connection connection) //dbutils.java
...
310. connection.close();
```

[Table of Contents] [Description]

Path 32:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
305. public static void closeConnection(Connection connection) //dbutils.java
       ...
317. connection.close();
```

[Table of Contents] [Description]

Path 33:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
349. public static void closeStatement(Statement statement) //dbutils.java
       ...
351. if (statement != null) statement.close();
```

[Table of Contents] [Description]

Path 34:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
361. public static void closeResultSet(ResultSet rs) //dbutils.java
       ...
363. if (rs != null) rs.close();
```

[Table of Contents] [Description]

Path 35:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
229. public static void closeConnection(Connection connection) //dbutils2.java
       ...
234. connection.close();
```

[Table of Contents] [Description]

Path 36:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
229. public static void closeConnection(Connection connection) //dbutils2.java
       ...
241. connection.close();
```

[Table of Contents] [Description]

Path 37:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
       ...
198. writer.close();
```

[Table of Contents] [Description]

Path 38:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
       ...
201. stream.close();
```

[Table of Contents] [Description]

Path 39:

Query Name - **Unclosed_Objects**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
354. if (transport != null) transport.close();
```

[Table of Contents] [Description]

Catch Without General Exception Handling

CWE ID	10009
Description	Not handling general exceptions may cause the program to crash.
Likelihood of Exploit	Low
Common Consequences	Program crashing.
Potential Mitigations	Handle the case of general exception.

Demonstrative Examples	The following code catches specific exception but not the general one.
	
	...
	try
	{
	File.Copy(path, path2);
	}
	catch(IOException e)
	{
	Console.WriteLine(e.Message);
	}
	...

Applicable Platforms	All

Path 1:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
198. public final static void waitln() //utility.java
       ...
200. try {
```

[Table of Contents] [Description]

Path 2:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
46. public String getMD5_Base64(String input) //mvntomcatjdbcrealm.java
       ...
60. try {
```

[Table of Contents] [Description]

Path 3:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
       ...
572. try {
```

[Table of Contents] [Description]

Path 4:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
       ...
588. try {
```

[Table of Contents] [Description]

Path 5:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
       ...
601. try {
```

[Table of Contents] [Description]

Path 6:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
659. public static void migratePrivateMessages(PrintStream ostream) //migrator.java
     ...
694. try {
```

[Table of Contents] [Description]

Path 7:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
659. public static void migratePrivateMessages(PrintStream ostream) //migrator.java
     ...
700. try {
```

[Table of Contents] [Description]

Path 8:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
160. public void convert() //phpbbtomvnforum.java
     161. try {
```

[Table of Contents] [Description]

Path 9:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
160. public void convert() //phpbbtomvnforum.java
     ...
188. try {
```

[Table of Contents] [Description]

Path 10:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
102. public void convert() //runningversion.java
     ...
133. try {
```

[Table of Contents] [Description]

Path 11:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
59. public void findByPrimaryKey(int categoryID) //categorydaoimpljdbc.java
      ...
      69. try {
```

[Table of Contents] [Description]

Path 12:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void findByAlternateKey_CategoryName(String
categoryName) //categorydaoimpljdbc.java
      ...
      97. try {
```

[Table of Contents] [Description]

Path 13:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
120. public void create(int categoryID, int parentCategoryID, String
categoryName, //categorydaoimpljdbc.java
      ...
      128. try {
```

[Table of Contents] [Description]

Path 14:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
120. public void create(int categoryID, int parentCategoryID, String
categoryName, //categorydaoimpljdbc.java
      ...
      139. try {
```

[Table of Contents] [Description]

Path 15:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
120. public void create(int categoryID, int parentCategoryID, String
categoryName, //categorydaoimpljdbc.java
      ...
      153. try {
```

[Table of Contents] [Description]

Path 16:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
59. public void findByPrimaryKey(int forumID) //forumdaoimpljdbc.java
      ...
69. try {
```

[Table of Contents] [Description]

Path 17:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
86. public void findByAlternateKey_ForumName_CategoryID(String forumName, int
categoryID) //forumdaoimpljdbc.java
      ...
96. try {
```

[Table of Contents] [Description]

Path 18:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int forumID, int categoryID, String
lastPostMemberName, //forumdaoimpljdbc.java
      ...
134. try {
```

[Table of Contents] [Description]

Path 19:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int forumID, int categoryID, String
lastPostMemberName, //forumdaoimpljdbc.java
      ...
145. try {
```

[Table of Contents] [Description]

Path 20:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int forumID, int categoryID, String
lastPostMemberName, //forumdaoimpljdbc.java
```

```
...
159. try {
```

[Table of Contents] [Description]

Path 21:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public void findByPrimaryKey(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
...
70. try {
```

[Table of Contents] [Description]

Path 22:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
...
91. try {
```

[Table of Contents] [Description]

Path 23:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
...
107. try {
```

[Table of Contents] [Description]

Path 24:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void findByPrimaryKey(int groupID) //groupsdaoimpljdbc.java
...
73. try {
```

[Table of Contents] [Description]

Path 25:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void findByAlternateKey_GroupName(String
groupName) //groupsdaoimpljdbc.java
      ...
      100. try {
```

[Table of Contents] [Description]

Path 26:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
      ...
      128. try {
```

[Table of Contents] [Description]

Path 27:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
      ...
      137. try {
```

[Table of Contents] [Description]

Path 28:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
      ...
      143. try {
```

[Table of Contents] [Description]

Path 29:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
      ...
      160. try {
```

[Table of Contents] [Description]

Path 30:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
61. public void findByPrimaryKey(int memberID) //memberdaoimpl.java
      ...
73. try {
```

[Table of Contents] [Description]

Path 31:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
96. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpl.java
      ...
108. try {
```

[Table of Contents] [Description]

Path 32:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
127. public void findByAlternateKey_MemberName(String
memberName) //memberdaoimpl.java
      ...
139. try {
```

[Table of Contents] [Description]

Path 33:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
      ...
199. try {
```

[Table of Contents] [Description]

Path 34:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
```

```
memberPassword, //memberdaoimpl.java
    ...
210. try {
```

[Table of Contents] [Description]

Path 35:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
    ...
221. try {
```

[Table of Contents] [Description]

Path 36:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
    ...
235. try {
```

[Table of Contents] [Description]

Path 37:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
317. public MemberBean getBean(int memberID) //memberdaoimpl.java
    ...
327. try {
```

[Table of Contents] [Description]

Path 38:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpl.java
    ...
439. try {
```

[Table of Contents] [Description]

Path 39:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpl.java
       ...
452. try {
```

[Table of Contents] [Description]

Path 40:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
523. public String getMemberNameFromMemberID(int memberID) //memberdaoimpl.java
       ...
533. try {
```

[Table of Contents] [Description]

Path 41:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public void findByPrimaryKey(int memberID) //memberdaoimpljdbc.java
       ...
72. try {
```

[Table of Contents] [Description]

Path 42:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
       ...
106. try {
```

[Table of Contents] [Description]

Path 43:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
125. public void findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
       ...
137. try {
```

[Table of Contents] [Description]

Path 44:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
    ...
197. try {
```

[Table of Contents] [Description]

Path 45:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
    ...
208. try {
```

[Table of Contents] [Description]

Path 46:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
    ...
219. try {
```

[Table of Contents] [Description]

Path 47:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
    ...
233. try {
```

[Table of Contents] [Description]

Path 48:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
316. public MemberBean getBean(int memberID) //memberdaoimpljdbc.java
    ...
327. try {
```

[Table of Contents] [Description]

Path 49:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpljdbc.java
      ...
659. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpljdbc.java
      ...
452. try {
```

[Table of Contents] [Description]

Path 51:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
522. public String getMemberNameFromMemberID(int memberID) //memberdaoimpljdbc.java
      ...
532. try {
```

[Table of Contents] [Description]

Path 52:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
550. public MemberBean getMemberFromMemberName(String memberName) throws
ObjectNotFoundException, DatabaseException //memberdaoimpljdbc.java
      ...
559. try {
```

[Table of Contents] [Description]

Path 53:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int
memberEmailVisible, //memberdaoimpljdbc.java
      |
```

```
...
643. try {
```

[Table of Contents] [Description]

Path 54:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int
memberEmailVisible, //memberdaoimpljdbc.java
...
654. try {
```

[Table of Contents] [Description]

Path 55:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int
memberEmailVisible, //memberdaoimpljdbc.java
...
668. try {
```

[Table of Contents] [Description]

Path 56:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public void findByPrimaryKey(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
70. try {
```

[Table of Contents] [Description]

Path 57:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
90. try {
```

[Table of Contents] [Description]

Path 58:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
101. try {
```

[Table of Contents] [Description]

Path 59:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
117. try {
```

[Table of Contents] [Description]

Path 60:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public void findByPrimaryKey(int messageID) //messagedaoimpljdbc.java
      ...
68. try {
```

[Table of Contents] [Description]

Path 61:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
92. public int create(String folderName, int memberID, int messageSenderID, String
messageSenderName, //messagedaoimpljdbc.java
      ...
106. try {
```

[Table of Contents] [Description]

Path 62:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void findByPrimaryKey(String folderName, int
memberID) //messagefolderdaoimpljdbc.java
      ...
```

```
73. try {
```

[Table of Contents] [Description]

Path 63:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public void create(String folderName, int memberID, int folderOrder, int
folderStatus, int folderOption, //messagefolderdaoimpljdbc.java
...
96. try {
```

[Table of Contents] [Description]

Path 64:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public void create(String folderName, int memberID, int folderOrder, int
folderStatus, int folderOption, //messagefolderdaoimpljdbc.java
...
110. try {
```

[Table of Contents] [Description]

Path 65:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
57. public Collection getBeans() //phpbbauthaccessdaoimpl.java
...
68. try {
```

[Table of Contents] [Description]

Path 66:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
100. public Collection getBeansByForumID(int forumId) throws
DatabaseException //phpbbauthaccessdaoimpl.java
...
110. try {
```

[Table of Contents] [Description]

Path 67:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public Collection getBeans() //phpbbcategoriesdaoimpl.java
      |...
      75. try {
```

[Table of Contents] [Description]

Path 68:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public Collection getBeans() throws DatabaseException //phpbbforumdaoimpl.java
      |...
      76. try {
```

[Table of Contents] [Description]

Path 69:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
116. public Collection getBeansByCategoryID(int categoryID) throws
DatabaseException //phpbbforumdaoimpl.java
      |...
      127. try {
```

[Table of Contents] [Description]

Path 70:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public Collection getBeans() //phpbbgroupdaoimpl.java
      |...
      69. try {
```

[Table of Contents] [Description]

Path 71:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public PhpbbGroup getBean(int groupID) throws ObjectNotFoundException,
DatabaseException //phpbbgroupdaoimpl.java
      |...
      101. try {
```

[Table of Contents] [Description]

Path 72:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
127. public PhpbbGroup getTruthGroup(int groupID) //phpbbgroupdaoimpl.java
       ...
136. try {
```

[Table of Contents] [Description]

Path 73:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
64. public int getPosterIDFromPostID(int post_id) //phpbbpostdaoimpl.java
       ...
75. try {
```

[Table of Contents] [Description]

Path 74:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
102. public Collection getBeans() //phpbbpostdaoimpl.java
       ...
115. try {
```

[Table of Contents] [Description]

Path 75:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
152. public Collection getPostIDsFromTopicID(int topicID) //phpbbpostdaoimpl.java
       ...
163. try {
```

[Table of Contents] [Description]

Path 76:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
185. public long getPostTimeFromPostID(int post_id) throws ObjectNotFoundException,
DatabaseException //phpbbpostdaoimpl.java
       ...
194. try {
```

[Table of Contents] [Description]

Path 77:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public Collection getBeansByThreadID(int threadID) //phpbbpostdaoimpl.java
       ...
227. try {
```

[Table of Contents] [Description]

Path 78:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public PhpbbPostText getBean(int post_id) throws ObjectNotFoundException,
DatabaseException //phpbbposttextdaoimpl.java
       ...
69. try {
```

[Table of Contents] [Description]

Path 79:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
97. public String getPostTextFromPostID(int post_id) throws ObjectNotFoundException,
DatabaseException //phpbbposttextdaoimpl.java
       ...
106. try {
```

[Table of Contents] [Description]

Path 80:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public Collection getBeans() //phpbbprivmmsgsdaoimpl.java
       ...
76. try {
```

[Table of Contents] [Description]

Path 81:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public Collection getBeansBySenderID(int senderID) //phpbbprivmmsgsdaoimpl.java
       ...
118. try {
```

[Table of Contents] [Description]

Path 82:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
148. public Collection getBeansByReceiverID(int
receiverID) //phpbbprivmmsgsdaoimpl.java
      ...
161. try {
```

[Table of Contents] [Description]

Path 83:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
191. public Collection getBeansByType(int privmsgsType, int
memberID) //phpbbprivmmsgsdaoimpl.java
      ...
204. try {
```

[Table of Contents] [Description]

Path 84:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
235. public Collection getBeansByTypeAndReceiveUser(int privmsgsType, int
memberID) //phpbbprivmmsgsdaoimpl.java
      ...
248. try {
```

[Table of Contents] [Description]

Path 85:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public PhpbbPrivmMsgsText getBean(int privmsgs_text_id) throws
ObjectNotFoundException, DatabaseException //phpbbprivmmsgstextdaoimpl.java
      ...
69. try {
```

[Table of Contents] [Description]

Path 86:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
57. public Collection getBeans() throws DatabaseException //phpbbranksdaoimpl.java
      ...
```

```
67. try {
```

[Table of Contents] [Description]

Path 87:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
64. public Collection getBeans() throws DatabaseException //phpbbtopicsdaoimpl.java
    ...
75. try {
```

[Table of Contents] [Description]

Path 88:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public Collection getBeansByForumID(int forumID) //phpbbtopicsdaoimpl.java
    ...
118. try {
```

[Table of Contents] [Description]

Path 89:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
150. public PhpbbTopics getBeans(int topicID) throws ObjectNotFoundException,
DatabaseException //phpbbtopicsdaoimpl.java
    ...
159. try {
```

[Table of Contents] [Description]

Path 90:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
57. public Collection getBeansByUserID(int userID) //phpbbtopicswatchdaoimpl.java
    ...
69. try {
```

[Table of Contents] [Description]

Path 91:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
91. public Collection getBeans(int userID, int
```

```
topicID) //phpbbtopicswatchdaoimpl.java
    ...
102. try {
```

[Table of Contents] [Description]

Path 92:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
127. public Collection getBeansByTopicID(int topicID) throws
DatabaseException //phpbbtopicswatchdaoimpl.java
    ...
137. try {
```

[Table of Contents] [Description]

Path 93:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public String getUserNameFromUserID (int userID) throws ObjectNotFoundException,
DatabaseException //phpbbuserdaoimpl.java
    ...
73. try {
```

[Table of Contents] [Description]

Path 94:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public Collection getBeans() throws DatabaseException //phpbbuserdaoimpl.java
    ...
117. try {
```

[Table of Contents] [Description]

Path 95:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
57. public Collection getBeansByGroupID(int groupID) throws
DatabaseException //phpbbusergroupdaoimpl.java
    ...
68. try {
```

[Table of Contents] [Description]

Path 96:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public void findByPrimaryKey(int postID) //postdaoimpljdbc.java
      ...
70. try {
```

[Table of Contents] [Description]

Path 97:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void create(int postID, int parentPostID, int
forumID, //postdaoimpljdbc.java
      ...
106. try {
```

[Table of Contents] [Description]

Path 98:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void create(int postID, int parentPostID, int
forumID, //postdaoimpljdbc.java
      ...
119. try {
```

[Table of Contents] [Description]

Path 99:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
61. public void findByAlternateKey_RankTitle(String
rankTitle) //rankdaoimpljdbc.java
      ...
71. try {
```

[Table of Contents] [Description]

Path 100:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void findByAlternateKey_RankMinPosts(int
rankMinPosts) //rankdaoimpljdbc.java
      ...
106. try {
```

[Table of Contents] [Description]

Path 101:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
     ...
128. try {
```

[Table of Contents] [Description]

Path 102:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
     ...
139. try {
```

[Table of Contents] [Description]

Path 103:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
     ...
153. try {
```

[Table of Contents] [Description]

Path 104:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
192. public Collection getRanks() //rankdaoimpljdbc.java
     ...
202. try {
```

[Table of Contents] [Description]

Path 105:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
59. public void findByPrimaryKey(int threadID) //threaddaoimpljdbc.java
     ...
```

```
69. try {
```

[Table of Contents] [Description]

Path 106:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public int createThread(int threadID, int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
103. try {
```

[Table of Contents] [Description]

Path 107:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public int createThread(int threadID, int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
121. try {
```

[Table of Contents] [Description]

Path 108:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public void findByPrimaryKey(int watchID) //watchdaoimpljdbc.java
    ...
74. try {
```

[Table of Contents] [Description]

Path 109:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
91. public void findByAlternateKey_MemberID_CategoryID_ForumID_ThreadID(int
memberID, int categoryID, int forumID, //watchdaoimpljdbc.java
    ...
101. try {
```

[Table of Contents] [Description]

Path 110:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
    ...
145. try {
```

[Table of Contents] [Description]

Path 111:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
    ...
154. try {
```

[Table of Contents] [Description]

Path 112:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
    ...
162. try {
```

[Table of Contents] [Description]

Path 113:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
    ...
172. try {
```

[Table of Contents] [Description]

Path 114:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
    ...
182. try {
```

[Table of Contents] [Description]

Path 115:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
      ...
197. try {
```

[Table of Contents] [Description]

Path 116:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public class DBUtils //dbutils.java
      ...
72. try {
```

[Table of Contents] [Description]

Path 117:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
123. public static void closeResultSet(ResultSet rs) //dbutils.java
      ...
125. try {
```

[Table of Contents] [Description]

Path 118:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public static void closeStatement(Statement statement) //dbutils.java
      ...
139. try {
```

[Table of Contents] [Description]

Path 119:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
153. public static void closeConnection(Connection connection) //dbutils.java
      ...
155. try {
```

[Table of Contents] [Description]

Path 120:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
167. public static void writeXmlFile(Document doc, String filename) //dbutils.java
     168. try {
```

[Table of Contents] [Description]

Path 121:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
186. public static FileInputStream getInput() //dbutils.java
     187. try {
```

[Table of Contents] [Description]

Path 122:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
196. public static FileOutputStream getOutput() //dbutils.java
     197. try {
```

[Table of Contents] [Description]

Path 123:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
125. public static synchronized String getMD5_Base64(String input) //md5.java
     ...
     141. try {
```

[Table of Contents] [Description]

Path 124:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
162. public static void main(String[] args) //md5.java
     ...
     169. try {
```

[Table of Contents] [Description]

Path 125:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
162. public static void main(String[] args) //md5.java
       ...
       175. try {
```

[Table of Contents] [Description]

Path 126:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
       ...
       73. try {
```

[Table of Contents] [Description]

Path 127:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
       ...
       93. try {
```

[Table of Contents] [Description]

Path 128:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
       ...
       329. try {
```

[Table of Contents] [Description]

Path 129:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
       ...
       341. try {
```

[Table of Contents] [Description]

Path 130:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1051. public static void main(String[] args) //importexportxml.java
      ...
1054. try {
```

[Table of Contents] [Description]

Path 131:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
10. public class XMLUtil //xmlutil.java
      ...
19. try {
```

[Table of Contents] [Description]

Path 132:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
248. public void handleText(final java.lang.String value) //attachmentlistimpl.java
      ...
254. try {
```

[Table of Contents] [Description]

Path 133:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
142. try {
```

[Table of Contents] [Description]

Path 134:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
152. try {
```

[Table of Contents] [Description]

Path 135:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
161. try {
```

[Table of Contents] [Description]

Path 136:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
170. try {
```

[Table of Contents] [Description]

Path 137:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
180. try {
```

[Table of Contents] [Description]

Path 138:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
      ...
191. try {
```

[Table of Contents] [Description]

Path 139:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
```

```
...
202. try {
```

[Table of Contents] [Description]

Path 140:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
...
213. try {
```

[Table of Contents] [Description]

Path 141:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
...
224. try {
```

[Table of Contents] [Description]

Path 142:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
...
235. try {
```

[Table of Contents] [Description]

Path 143:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
132. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //attachmenttypeimpl.java
...
246. try {
```

[Table of Contents] [Description]

Path 144:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
678. public void handleText(final java.lang.String value) //attachmenttypeimpl.java
       ...
684. try {
```

[Table of Contents] [Description]

Path 145:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
765. private void eatText1(final java.lang.String value) //attachmenttypeimpl.java
       ...
768. try {
```

[Table of Contents] [Description]

Path 146:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
775. private void eatText2(final java.lang.String value) //attachmenttypeimpl.java
       ...
778. try {
```

[Table of Contents] [Description]

Path 147:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
785. private void eatText3(final java.lang.String value) //attachmenttypeimpl.java
       ...
788. try {
```

[Table of Contents] [Description]

Path 148:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
796. private void eatText4(final java.lang.String value) //attachmenttypeimpl.java
       ...
799. try {
```

[Table of Contents] [Description]

Path 149:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
807. private void eatText5(final java.lang.String value) //attachmenttypeimpl.java
      ...
810. try {
```

[Table of Contents] [Description]

Path 150:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
818. private void eatText6(final java.lang.String value) //attachmenttypeimpl.java
      ...
821. try {
```

[Table of Contents] [Description]

Path 151:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
828. private void eatText7(final java.lang.String value) //attachmenttypeimpl.java
      ...
831. try {
```

[Table of Contents] [Description]

Path 152:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
838. private void eatText8(final java.lang.String value) //attachmenttypeimpl.java
      ...
841. try {
```

[Table of Contents] [Description]

Path 153:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
848. private void eatText9(final java.lang.String value) //attachmenttypeimpl.java
      ...
851. try {
```

[Table of Contents] [Description]

Path 154:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
859. private void eatText10(final java.lang.String value) //attachmenttypeimpl.java
     ...
862. try {
```

[Table of Contents] [Description]

Path 155:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
869. private void eatText11(final java.lang.String value) //attachmenttypeimpl.java
     ...
872. try {
```

[Table of Contents] [Description]

Path 156:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //categorylisttypeimpl.java
     ...
250. try {
```

[Table of Contents] [Description]

Path 157:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
127. try {
```

[Table of Contents] [Description]

Path 158:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
137. try {
```

[Table of Contents] [Description]

Path 159:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
148. try {
```

[Table of Contents] [Description]

Path 160:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
159. try {
```

[Table of Contents] [Description]

Path 161:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
170. try {
```

[Table of Contents] [Description]

Path 162:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorytypeimpl.java
     ...
181. try {
```

[Table of Contents] [Description]

Path 163:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
```

```
context) //categorytypeimpl.java
    ...
192. try {
```

[Table of Contents] [Description]

Path 164:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
698. public void handleText(final java.lang.String value) //categorytypeimpl.java
    ...
704. try {
```

[Table of Contents] [Description]

Path 165:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
781. private void eatText1(final java.lang.String value) //categorytypeimpl.java
    ...
784. try {
```

[Table of Contents] [Description]

Path 166:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
791. private void eatText2(final java.lang.String value) //categorytypeimpl.java
    ...
794. try {
```

[Table of Contents] [Description]

Path 167:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
802. private void eatText3(final java.lang.String value) //categorytypeimpl.java
    ...
805. try {
```

[Table of Contents] [Description]

Path 168:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
813. private void eatText4(final java.lang.String value) //categorytypeimpl.java
      ...
816. try {
```

[Table of Contents] [Description]

Path 169:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
823. private void eatText5(final java.lang.String value) //categorytypeimpl.java
      ...
826. try {
```

[Table of Contents] [Description]

Path 170:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
833. private void eatText6(final java.lang.String value) //categorytypeimpl.java
      ...
836. try {
```

[Table of Contents] [Description]

Path 171:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
843. private void eatText7(final java.lang.String value) //categorytypeimpl.java
      ...
846. try {
```

[Table of Contents] [Description]

Path 172:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorywatchlistimpl.java
      ...
45. try {
```

[Table of Contents] [Description]

Path 173:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorywatchlistimpl.java
      ...
      52. try {
```

[Table of Contents] [Description]

Path 174:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorywatchlistimpl.java
      ...
      58. try {
```

[Table of Contents] [Description]

Path 175:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public void serializeAttributes(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorywatchlistimpl.java
      ...
      73. try {
```

[Table of Contents] [Description]

Path 176:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void serializeURIs(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //categorywatchlistimpl.java
      ...
      87. try {
```

[Table of Contents] [Description]

Path 177:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
256. public void handleText(final java.lang.String
value) //categorywatchlistimpl.java
      ...
      262. try {
```

[Table of Contents] [Description]

Path 178:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
282. private void eatText1(final java.lang.String
value) //categorywatchlistimpl.java
      ...
285. try {
```

[Table of Contents] [Description]

Path 179:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //favoritethreadlistimpl.java
      ...
250. try {
```

[Table of Contents] [Description]

Path 180:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //favoritethreadtypeimpl.java
      ...
82. try {
```

[Table of Contents] [Description]

Path 181:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //favoritethreadtypeimpl.java
      ...
92. try {
```

[Table of Contents] [Description]

Path 182:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //favoritethreadtypeimpl.java
    ...
103. try {
```

[Table of Contents] [Description]

Path 183:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //favoritethreadtypeimpl.java
    ...
114. try {
```

[Table of Contents] [Description]

Path 184:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //favoritethreadtypeimpl.java
    ...
125. try {
```

[Table of Contents] [Description]

Path 185:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
415. public void handleText(final java.lang.String
value) //favoritethreadtypeimpl.java
    ...
421. try {
```

[Table of Contents] [Description]

Path 186:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
466. private void eatText1(final java.lang.String
value) //favoritethreadtypeimpl.java
    ...
469. try {
```

[Table of Contents] [Description]

Path 187:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
477. private void eatText2(final java.lang.String
value) //favoritethreadtypeimpl.java
      |...
      |480. try {
```

[Table of Contents] [Description]

Path 188:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
488. private void eatText3(final java.lang.String
value) //favoritethreadtypeimpl.java
      |...
      |491. try {
```

[Table of Contents] [Description]

Path 189:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
498. private void eatText4(final java.lang.String
value) //favoritethreadtypeimpl.java
      |...
      |501. try {
```

[Table of Contents] [Description]

Path 190:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
509. private void eatText5(final java.lang.String
value) //favoritethreadtypeimpl.java
      |...
      |512. try {
```

[Table of Contents] [Description]

Path 191:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
247. public void handleText(final java.lang.String value) //forumlistimpl.java
      |
```

```
...
253. try {
```

[Table of Contents] [Description]

Path 192:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
219. try {
```

[Table of Contents] [Description]

Path 193:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
229. try {
```

[Table of Contents] [Description]

Path 194:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
239. try {
```

[Table of Contents] [Description]

Path 195:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
250. try {
```

[Table of Contents] [Description]

Path 196:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      261. try {
```

[Table of Contents] [Description]

Path 197:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      272. try {
```

[Table of Contents] [Description]

Path 198:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      283. try {
```

[Table of Contents] [Description]

Path 199:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      294. try {
```

[Table of Contents] [Description]

Path 200:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
```

```
305. try {
```

[Table of Contents] [Description]

Path 201:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
316. try {
```

[Table of Contents] [Description]

Path 202:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
327. try {
```

[Table of Contents] [Description]

Path 203:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
338. try {
```

[Table of Contents] [Description]

Path 204:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
...
349. try {
```

[Table of Contents] [Description]

Path 205:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      360. try {
```

[Table of Contents] [Description]

Path 206:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumtypeimpl.java
      ...
      371. try {
```

[Table of Contents] [Description]

Path 207:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1140. public void handleText(final java.lang.String value) //forumtypeimpl.java
      ...
      1146. try {
```

[Table of Contents] [Description]

Path 208:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1285. private void eatText1(final java.lang.String value) //forumtypeimpl.java
      ...
      1288. try {
```

[Table of Contents] [Description]

Path 209:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1295. private void eatText2(final java.lang.String value) //forumtypeimpl.java
      ...
      1298. try {
```

[Table of Contents] [Description]

Path 210:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1305. private void eatText3(final java.lang.String value) //forumtypeimpl.java
      ...
1308. try {
```

[Table of Contents] [Description]

Path 211:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1315. private void eatText4(final java.lang.String value) //forumtypeimpl.java
      ...
1318. try {
```

[Table of Contents] [Description]

Path 212:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1326. private void eatText5(final java.lang.String value) //forumtypeimpl.java
      ...
1329. try {
```

[Table of Contents] [Description]

Path 213:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1337. private void eatText6(final java.lang.String value) //forumtypeimpl.java
      ...
1340. try {
```

[Table of Contents] [Description]

Path 214:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1347. private void eatText7(final java.lang.String value) //forumtypeimpl.java
      ...
1350. try {
```

[Table of Contents] [Description]

Path 215:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1358. private void eatText8(final java.lang.String value) //forumtypeimpl.java
      ...
1361. try {
```

[Table of Contents] [Description]

Path 216:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1369. private void eatText9(final java.lang.String value) //forumtypeimpl.java
      ...
1372. try {
```

[Table of Contents] [Description]

Path 217:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1379. private void eatText10(final java.lang.String value) //forumtypeimpl.java
      ...
1382. try {
```

[Table of Contents] [Description]

Path 218:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1390. private void eatText11(final java.lang.String value) //forumtypeimpl.java
      ...
1393. try {
```

[Table of Contents] [Description]

Path 219:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1400. private void eatText12(final java.lang.String value) //forumtypeimpl.java
      ...
1403. try {
```

[Table of Contents] [Description]

Path 220:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1410. private void eatText13(final java.lang.String value) //forumtypeimpl.java
      ...
1413. try {
```

[Table of Contents] [Description]

Path 221:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1421. private void eatText14(final java.lang.String value) //forumtypeimpl.java
      ...
1424. try {
```

[Table of Contents] [Description]

Path 222:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1432. private void eatText15(final java.lang.String value) //forumtypeimpl.java
      ...
1435. try {
```

[Table of Contents] [Description]

Path 223:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumwatchlistimpl.java
      ...
45. try {
```

[Table of Contents] [Description]

Path 224:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumwatchlistimpl.java
      ...
52. try {
```

[Table of Contents] [Description]

Path 225:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumwatchlistimpl.java
      ...
58. try {
```

[Table of Contents] [Description]

Path 226:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public void serializeAttributes(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumwatchlistimpl.java
      ...
73. try {
```

[Table of Contents] [Description]

Path 227:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void serializeURIs(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //forumwatchlistimpl.java
      ...
87. try {
```

[Table of Contents] [Description]

Path 228:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
256. public void handleText(final java.lang.String value) //forumwatchlistimpl.java
      ...
262. try {
```

[Table of Contents] [Description]

Path 229:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
282. private void eatText1(final java.lang.String value) //forumwatchlistimpl.java
      |
```

```
...
285. try {
```

[Table of Contents] [Description]

Path 230:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalpermissionlistimpl.java
...
45. try {
```

[Table of Contents] [Description]

Path 231:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalpermissionlistimpl.java
...
52. try {
```

[Table of Contents] [Description]

Path 232:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
37. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalpermissionlistimpl.java
...
58. try {
```

[Table of Contents] [Description]

Path 233:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public void serializeAttributes(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalpermissionlistimpl.java
...
73. try {
```

[Table of Contents] [Description]

Path 234:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void serializeURIs(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalpermissionlistimpl.java
      ...
87. try {
```

[Table of Contents] [Description]

Path 235:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
275. public void handleText(final java.lang.String
value) //globalpermissionlistimpl.java
      ...
281. try {
```

[Table of Contents] [Description]

Path 236:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
301. private void eatText1(final java.lang.String
value) //globalpermissionlistimpl.java
      ...
304. try {
```

[Table of Contents] [Description]

Path 237:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
274. public void handleText(final java.lang.String value) //globalwatchlistimpl.java
      ...
280. try {
```

[Table of Contents] [Description]

Path 238:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
92. try {
```

[Table of Contents] [Description]

Path 239:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
103. try {
```

[Table of Contents] [Description]

Path 240:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
114. try {
```

[Table of Contents] [Description]

Path 241:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
125. try {
```

[Table of Contents] [Description]

Path 242:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
136. try {
```

[Table of Contents] [Description]

Path 243:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
85. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //globalwatchtypeimpl.java
      ...
147. try {
```

[Table of Contents] [Description]

Path 244:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
473. public void handleText(final java.lang.String value) //globalwatchtypeimpl.java
      ...
479. try {
```

[Table of Contents] [Description]

Path 245:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
534. private void eatText1(final java.lang.String value) //globalwatchtypeimpl.java
      ...
537. try {
```

[Table of Contents] [Description]

Path 246:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
544. private void eatText2(final java.lang.String value) //globalwatchtypeimpl.java
      ...
547. try {
```

[Table of Contents] [Description]

Path 247:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
555. private void eatText3(final java.lang.String value) //globalwatchtypeimpl.java
      ...
558. try {
```

[Table of Contents] [Description]

Path 248:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
565. private void eatText4(final java.lang.String value) //globalwatchtypeimpl.java
     ...
568. try {
```

[Table of Contents] [Description]

Path 249:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
575. private void eatText5(final java.lang.String value) //globalwatchtypeimpl.java
     ...
578. try {
```

[Table of Contents] [Description]

Path 250:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
586. private void eatText6(final java.lang.String value) //globalwatchtypeimpl.java
     ...
589. try {
```

[Table of Contents] [Description]

Path 251:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //groupforumpermissionlistimpl.java
     ...
250. try {
```

[Table of Contents] [Description]

Path 252:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupforumpermissiontypeimpl.java
     ...
54. try {
```

[Table of Contents] [Description]

Path 253:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupforumpermissiontypeimpl.java
      |...
      |63. try {
```

[Table of Contents] [Description]

Path 254:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. public void handleText(final java.lang.String
value) //groupforumpermissiontypeimpl.java
      |...
      |275. try {
```

[Table of Contents] [Description]

Path 255:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. private void eatText1(final java.lang.String
value) //groupforumpermissiontypeimpl.java
      |...
      |299. try {
```

[Table of Contents] [Description]

Path 256:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
306. private void eatText2(final java.lang.String
value) //groupforumpermissiontypeimpl.java
      |...
      |309. try {
```

[Table of Contents] [Description]

Path 257:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String value) //groupmemberlistimpl.java
      |...
```

```
250. try {
```

[Table of Contents] [Description]

Path 258:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupmembertypeimpl.java
    ...
69. try {
```

[Table of Contents] [Description]

Path 259:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupmembertypeimpl.java
    ...
79. try {
```

[Table of Contents] [Description]

Path 260:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupmembertypeimpl.java
    ...
90. try {
```

[Table of Contents] [Description]

Path 261:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //groupmembertypeimpl.java
    ...
101. try {
```

[Table of Contents] [Description]

Path 262:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
364. public void handleText(final java.lang.String value) //groupmembertypeimpl.java
     ...
370. try {
```

[Table of Contents] [Description]

Path 263:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
408. private void eatText1(final java.lang.String value) //groupmembertypeimpl.java
     ...
411. try {
```

[Table of Contents] [Description]

Path 264:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
418. private void eatText2(final java.lang.String value) //groupmembertypeimpl.java
     ...
421. try {
```

[Table of Contents] [Description]

Path 265:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
428. private void eatText3(final java.lang.String value) //groupmembertypeimpl.java
     ...
431. try {
```

[Table of Contents] [Description]

Path 266:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
438. private void eatText4(final java.lang.String value) //groupmembertypeimpl.java
     ...
441. try {
```

[Table of Contents] [Description]

Path 267:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
    ...
106. try {
```

[Table of Contents] [Description]

Path 268:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
    ...
116. try {
```

[Table of Contents] [Description]

Path 269:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
    ...
126. try {
```

[Table of Contents] [Description]

Path 270:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
    ...
137. try {
```

[Table of Contents] [Description]

Path 271:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
    ...
148. try {
```

[Table of Contents] [Description]

Path 272:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //grouptypeimpl.java
      |...
      |159. try {
```

[Table of Contents] [Description]

Path 273:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
586. public void handleText(final java.lang.String value) //grouptypeimpl.java
      |...
      |592. try {
```

[Table of Contents] [Description]

Path 274:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
656. private void eatText1(final java.lang.String value) //grouptypeimpl.java
      |...
      |659. try {
```

[Table of Contents] [Description]

Path 275:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
666. private void eatText2(final java.lang.String value) //grouptypeimpl.java
      |...
      |669. try {
```

[Table of Contents] [Description]

Path 276:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
676. private void eatText3(final java.lang.String value) //grouptypeimpl.java
      |...
```

```
679. try {
```

[Table of Contents] [Description]

Path 277:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
687. private void eatText4(final java.lang.String value) //grouptypeimpl.java
     ...
690. try {
```

[Table of Contents] [Description]

Path 278:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
697. private void eatText5(final java.lang.String value) //grouptypeimpl.java
     ...
700. try {
```

[Table of Contents] [Description]

Path 279:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
707. private void eatText6(final java.lang.String value) //grouptypeimpl.java
     ...
710. try {
```

[Table of Contents] [Description]

Path 280:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //memberforumpermissionlistimpl.java
     ...
250. try {
```

[Table of Contents] [Description]

Path 281:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
```

```
context) //memberforumpermissiontypeimpl.java
    ...
54. try {
```

[Table of Contents] [Description]

Path 282:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //memberforumpermissiontypeimpl.java
    ...
63. try {
```

[Table of Contents] [Description]

Path 283:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. public void handleText(final java.lang.String
value) //memberforumpermissiontypeimpl.java
    ...
275. try {
```

[Table of Contents] [Description]

Path 284:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. private void eatText1(final java.lang.String
value) //memberforumpermissiontypeimpl.java
    ...
299. try {
```

[Table of Contents] [Description]

Path 285:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
306. private void eatText2(final java.lang.String
value) //memberforumpermissiontypeimpl.java
    ...
309. try {
```

[Table of Contents] [Description]

Path 286:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
547. try {
```

[Table of Contents] [Description]

Path 287:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
556. try {
```

[Table of Contents] [Description]

Path 288:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
565. try {
```

[Table of Contents] [Description]

Path 289:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
574. try {
```

[Table of Contents] [Description]

Path 290:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
```

```
...
584. try {
```

[Table of Contents] [Description]

Path 291:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
595. try {
```

[Table of Contents] [Description]

Path 292:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
606. try {
```

[Table of Contents] [Description]

Path 293:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
617. try {
```

[Table of Contents] [Description]

Path 294:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
628. try {
```

[Table of Contents] [Description]

Path 295:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
639. try {
```

[Table of Contents] [Description]

Path 296:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
650. try {
```

[Table of Contents] [Description]

Path 297:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
661. try {
```

[Table of Contents] [Description]

Path 298:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
672. try {
```

[Table of Contents] [Description]

Path 299:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
```

```
683. try {
```

[Table of Contents] [Description]

Path 300:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
694. try {
```

[Table of Contents] [Description]

Path 301:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
705. try {
```

[Table of Contents] [Description]

Path 302:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
716. try {
```

[Table of Contents] [Description]

Path 303:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
727. try {
```

[Table of Contents] [Description]

Path 304:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
738. try {
```

[Table of Contents] [Description]

Path 305:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
749. try {
```

[Table of Contents] [Description]

Path 306:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
760. try {
```

[Table of Contents] [Description]

Path 307:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
771. try {
```

[Table of Contents] [Description]

Path 308:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
782. try {
```

[Table of Contents] [Description]

Path 309:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
793. try {
```

[Table of Contents] [Description]

Path 310:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
804. try {
```

[Table of Contents] [Description]

Path 311:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
814. try {
```

[Table of Contents] [Description]

Path 312:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
824. try {
```

[Table of Contents] [Description]

Path 313:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
835. try {
```

[Table of Contents] [Description]

Path 314:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
846. try {
```

[Table of Contents] [Description]

Path 315:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
857. try {
```

[Table of Contents] [Description]

Path 316:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
868. try {
```

[Table of Contents] [Description]

Path 317:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
878. try {
```

[Table of Contents] [Description]

Path 318:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
887. try {
```

[Table of Contents] [Description]

Path 319:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
897. try {
```

[Table of Contents] [Description]

Path 320:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
908. try {
```

[Table of Contents] [Description]

Path 321:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
      ...
919. try {
```

[Table of Contents] [Description]

Path 322:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
```

```
context) //membertypeimpl.java
    ...
930. try {
```

[Table of Contents] [Description]

Path 323:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
941. try {
```

[Table of Contents] [Description]

Path 324:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
952. try {
```

[Table of Contents] [Description]

Path 325:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
963. try {
```

[Table of Contents] [Description]

Path 326:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
    ...
974. try {
```

[Table of Contents] [Description]

Path 327:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
985. try {
```

[Table of Contents] [Description]

Path 328:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
996. try {
```

[Table of Contents] [Description]

Path 329:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
1007. try {
```

[Table of Contents] [Description]

Path 330:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     ...
1018. try {
```

[Table of Contents] [Description]

Path 331:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
     |
```

```
...
1029. try {
```

[Table of Contents] [Description]

Path 332:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
1040. try {
```

[Table of Contents] [Description]

Path 333:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
1051. try {
```

[Table of Contents] [Description]

Path 334:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
1062. try {
```

[Table of Contents] [Description]

Path 335:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
541. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //membertypeimpl.java
...
1073. try {
```

[Table of Contents] [Description]

Path 336:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2697. public void handleText(final java.lang.String value) //membertypeimpl.java
      ...
2703. try {
```

[Table of Contents] [Description]

Path 337:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3069. private void eatText1(final java.lang.String value) //membertypeimpl.java
      ...
3072. try {
```

[Table of Contents] [Description]

Path 338:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3080. private void eatText2(final java.lang.String value) //membertypeimpl.java
      ...
3083. try {
```

[Table of Contents] [Description]

Path 339:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3090. private void eatText3(final java.lang.String value) //membertypeimpl.java
      ...
3093. try {
```

[Table of Contents] [Description]

Path 340:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3100. private void eatText4(final java.lang.String value) //membertypeimpl.java
      ...
3103. try {
```

[Table of Contents] [Description]

Path 341:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3111. private void eatText5(final java.lang.String value) //membertypeimpl.java
      ...
3114. try {
```

[Table of Contents] [Description]

Path 342:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3121. private void eatText6(final java.lang.String value) //membertypeimpl.java
      ...
3124. try {
```

[Table of Contents] [Description]

Path 343:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3131. private void eatText7(final java.lang.String value) //membertypeimpl.java
      ...
3134. try {
```

[Table of Contents] [Description]

Path 344:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3141. private void eatText8(final java.lang.String value) //membertypeimpl.java
      ...
3144. try {
```

[Table of Contents] [Description]

Path 345:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3151. private void eatText9(final java.lang.String value) //membertypeimpl.java
      ...
3154. try {
```

[Table of Contents] [Description]

Path 346:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3162. private void eatText10(final java.lang.String value) //membertypeimpl.java
      ...
3165. try {
```

[Table of Contents] [Description]

Path 347:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3173. private void eatText11(final java.lang.String value) //membertypeimpl.java
      ...
3176. try {
```

[Table of Contents] [Description]

Path 348:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3183. private void eatText12(final java.lang.String value) //membertypeimpl.java
      ...
3186. try {
```

[Table of Contents] [Description]

Path 349:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3193. private void eatText13(final java.lang.String value) //membertypeimpl.java
      ...
3196. try {
```

[Table of Contents] [Description]

Path 350:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3203. private void eatText14(final java.lang.String value) //membertypeimpl.java
      ...
3206. try {
```

[Table of Contents] [Description]

Path 351:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3213. private void eatText15(final java.lang.String value) //membertypeimpl.java
      ...
3216. try {
```

[Table of Contents] [Description]

Path 352:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3224. private void eatText16(final java.lang.String value) //membertypeimpl.java
      ...
3227. try {
```

[Table of Contents] [Description]

Path 353:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3235. private void eatText17(final java.lang.String value) //membertypeimpl.java
      ...
3238. try {
```

[Table of Contents] [Description]

Path 354:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3245. private void eatText18(final java.lang.String value) //membertypeimpl.java
      ...
3248. try {
```

[Table of Contents] [Description]

Path 355:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3255. private void eatText19(final java.lang.String value) //membertypeimpl.java
      ...
3258. try {
```

[Table of Contents] [Description]

Path 356:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3265. private void eatText20(final java.lang.String value) //membertypeimpl.java
      ...
3268. try {
```

[Table of Contents] [Description]

Path 357:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3276. private void eatText21(final java.lang.String value) //membertypeimpl.java
      ...
3279. try {
```

[Table of Contents] [Description]

Path 358:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3286. private void eatText22(final java.lang.String value) //membertypeimpl.java
      ...
3289. try {
```

[Table of Contents] [Description]

Path 359:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3296. private void eatText23(final java.lang.String value) //membertypeimpl.java
      ...
3299. try {
```

[Table of Contents] [Description]

Path 360:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3306. private void eatText24(final java.lang.String value) //membertypeimpl.java
      ...
3309. try {
```

[Table of Contents] [Description]

Path 361:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3316. private void eatText25(final java.lang.String value) //membertypeimpl.java
      ...
3319. try {
```

[Table of Contents] [Description]

Path 362:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3326. private void eatText26(final java.lang.String value) //membertypeimpl.java
      ...
3329. try {
```

[Table of Contents] [Description]

Path 363:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3336. private void eatText27(final java.lang.String value) //membertypeimpl.java
      ...
3339. try {
```

[Table of Contents] [Description]

Path 364:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3347. private void eatText28(final java.lang.String value) //membertypeimpl.java
      ...
3350. try {
```

[Table of Contents] [Description]

Path 365:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3358. private void eatText29(final java.lang.String value) //membertypeimpl.java
      ...
3361. try {
```

[Table of Contents] [Description]

Path 366:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3369. private void eatText30(final java.lang.String value) //membertypeimpl.java
      ...
3372. try {
```

[Table of Contents] [Description]

Path 367:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3379. private void eatText31(final java.lang.String value) //membertypeimpl.java
      ...
3382. try {
```

[Table of Contents] [Description]

Path 368:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3390. private void eatText32(final java.lang.String value) //membertypeimpl.java
      ...
3393. try {
```

[Table of Contents] [Description]

Path 369:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3400. private void eatText33(final java.lang.String value) //membertypeimpl.java
      ...
3403. try {
```

[Table of Contents] [Description]

Path 370:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3410. private void eatText34(final java.lang.String value) //membertypeimpl.java
      ...
3413. try {
```

[Table of Contents] [Description]

Path 371:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3420. private void eatText35(final java.lang.String value) //membertypeimpl.java
      ...
3423. try {
```

[Table of Contents] [Description]

Path 372:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3430. private void eatText36(final java.lang.String value) //membertypeimpl.java
      ...
3433. try {
```

[Table of Contents] [Description]

Path 373:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3440. private void eatText37(final java.lang.String value) //membertypeimpl.java
      ...
3443. try {
```

[Table of Contents] [Description]

Path 374:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3451. private void eatText38(final java.lang.String value) //membertypeimpl.java
      ...
3454. try {
```

[Table of Contents] [Description]

Path 375:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3461. private void eatText39(final java.lang.String value) //membertypeimpl.java
      ...
3464. try {
```

[Table of Contents] [Description]

Path 376:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3471. private void eatText40(final java.lang.String value) //membertypeimpl.java
      ...
3474. try {
```

[Table of Contents] [Description]

Path 377:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3481. private void eatText41(final java.lang.String value) //membertypeimpl.java
      ...
3484. try {
```

[Table of Contents] [Description]

Path 378:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3491. private void eatText42(final java.lang.String value) //membertypeimpl.java
      ...
3494. try {
```

[Table of Contents] [Description]

Path 379:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3501. private void eatText43(final java.lang.String value) //membertypeimpl.java
      ...
3504. try {
```

[Table of Contents] [Description]

Path 380:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3512. private void eatText44(final java.lang.String value) //membertypeimpl.java
      ...
3515. try {
```

[Table of Contents] [Description]

Path 381:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3522. private void eatText45(final java.lang.String value) //membertypeimpl.java
      ...
3525. try {
```

[Table of Contents] [Description]

Path 382:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3532. private void eatText46(final java.lang.String value) //membertypeimpl.java
      ...
3535. try {
```

[Table of Contents] [Description]

Path 383:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3542. private void eatText47(final java.lang.String value) //membertypeimpl.java
      ...
3545. try {
```

[Table of Contents] [Description]

Path 384:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3552. private void eatText48(final java.lang.String value) //membertypeimpl.java
      ...
3555. try {
```

[Table of Contents] [Description]

Path 385:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3563. private void eatText49(final java.lang.String value) //membertypeimpl.java
      ...
3566. try {
```

[Table of Contents] [Description]

Path 386:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
3573. private void eatText50(final java.lang.String value) //membertypeimpl.java
      ...
3576. try {
```

[Table of Contents] [Description]

Path 387:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //messagefolderlistimpl.java
      ...
250. try {
```

[Table of Contents] [Description]

Path 388:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagefoldertypeimpl.java
      ...
78. try {
```

[Table of Contents] [Description]

Path 389:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagefoldertypeimpl.java
      ...
88. try {
```

[Table of Contents] [Description]

Path 390:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagefoldertypeimpl.java
      ...
99. try {
```

[Table of Contents] [Description]

Path 391:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagefoldertypeimpl.java
      ...
110. try {
```

[Table of Contents] [Description]

Path 392:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
431. public void handleText(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
437. try {
```

[Table of Contents] [Description]

Path 393:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
481. private void eatText1(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
484. try {
```

[Table of Contents] [Description]

Path 394:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
491. private void eatText2(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
494. try {
```

[Table of Contents] [Description]

Path 395:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
501. private void eatText3(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
504. try {
```

[Table of Contents] [Description]

Path 396:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
511. private void eatText4(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
514. try {
```

[Table of Contents] [Description]

Path 397:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
313. public void handleText(final java.lang.String value) //messagelistimpl.java
      ...
319. try {
```

[Table of Contents] [Description]

Path 398:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
188. try {
```

[Table of Contents] [Description]

Path 399:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
199. try {
```

[Table of Contents] [Description]

Path 400:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
210. try {
```

[Table of Contents] [Description]

Path 401:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
221. try {
```

[Table of Contents] [Description]

Path 402:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
232. try {
```

[Table of Contents] [Description]

Path 403:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
      ...
243. try {
```

[Table of Contents] [Description]

Path 404:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
```

```
...
254. try {
```

[Table of Contents] [Description]

Path 405:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
...
265. try {
```

[Table of Contents] [Description]

Path 406:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
...
276. try {
```

[Table of Contents] [Description]

Path 407:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
...
287. try {
```

[Table of Contents] [Description]

Path 408:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
...
298. try {
```

[Table of Contents] [Description]

Path 409:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
     ...
309. try {
```

[Table of Contents] [Description]

Path 410:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
     ...
320. try {
```

[Table of Contents] [Description]

Path 411:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
     ...
331. try {
```

[Table of Contents] [Description]

Path 412:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
     ...
342. try {
```

[Table of Contents] [Description]

Path 413:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //messagetypeimpl.java
     ...
```

```
353. try {
```

[Table of Contents] [Description]

Path 414:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
941. public void handleText(final java.lang.String value) //messagetypeimpl.java
     ...
947. try {
```

[Table of Contents] [Description]

Path 415:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1072. private void eatText1(final java.lang.String value) //messagetypeimpl.java
      ...
1075. try {
```

[Table of Contents] [Description]

Path 416:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1083. private void eatText2(final java.lang.String value) //messagetypeimpl.java
      ...
1086. try {
```

[Table of Contents] [Description]

Path 417:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1093. private void eatText3(final java.lang.String value) //messagetypeimpl.java
      ...
1096. try {
```

[Table of Contents] [Description]

Path 418:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1104. private void eatText4(final java.lang.String value) //messagetypeimpl.java
```

```
...
1107. try {
```

[Table of Contents] [Description]

Path 419:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1114. private void eatText5(final java.lang.String value) //messagetypeimpl.java
...
1117. try {
```

[Table of Contents] [Description]

Path 420:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1124. private void eatText6(final java.lang.String value) //messagetypeimpl.java
...
1127. try {
```

[Table of Contents] [Description]

Path 421:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1134. private void eatText7(final java.lang.String value) //messagetypeimpl.java
...
1137. try {
```

[Table of Contents] [Description]

Path 422:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1145. private void eatText8(final java.lang.String value) //messagetypeimpl.java
...
1148. try {
```

[Table of Contents] [Description]

Path 423:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1155. private void eatText9(final java.lang.String value) //messagetypeimpl.java
```

```
...
1158. try {
```

[Table of Contents] [Description]

Path 424:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1165. private void eatText10(final java.lang.String value) //messagetypeimpl.java
...
1168. try {
```

[Table of Contents] [Description]

Path 425:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1175. private void eatText11(final java.lang.String value) //messagetypeimpl.java
...
1178. try {
```

[Table of Contents] [Description]

Path 426:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1186. private void eatText12(final java.lang.String value) //messagetypeimpl.java
...
1189. try {
```

[Table of Contents] [Description]

Path 427:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1197. private void eatText13(final java.lang.String value) //messagetypeimpl.java
...
1200. try {
```

[Table of Contents] [Description]

Path 428:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1208. private void eatText14(final java.lang.String value) //messagetypeimpl.java
```

```
...
1211. try {
```

[Table of Contents] [Description]

Path 429:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1218. private void eatText15(final java.lang.String value) //messagetypeimpl.java
...
1221. try {
```

[Table of Contents] [Description]

Path 430:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1228. private void eatText16(final java.lang.String value) //messagetypeimpl.java
...
1231. try {
```

[Table of Contents] [Description]

Path 431:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
246. public void handleText(final java.lang.String value) //mvnforumimpl.java
...
252. try {
```

[Table of Contents] [Description]

Path 432:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
412. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
...
418. try {
```

[Table of Contents] [Description]

Path 433:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
670. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
```

```
...
676. try {
```

[Table of Contents] [Description]

Path 434:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
928. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
...
934. try {
```

[Table of Contents] [Description]

Path 435:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1196. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
...
1202. try {
```

[Table of Contents] [Description]

Path 436:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
251. public void handleText(final java.lang.String value) //postlistimpl.java
...
257. try {
```

[Table of Contents] [Description]

Path 437:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
...
186. try {
```

[Table of Contents] [Description]

Path 438:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
197. try {
```

[Table of Contents] [Description]

Path 439:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
207. try {
```

[Table of Contents] [Description]

Path 440:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
216. try {
```

[Table of Contents] [Description]

Path 441:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
226. try {
```

[Table of Contents] [Description]

Path 442:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
237. try {
```

[Table of Contents] [Description]

Path 443:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
      ...
248. try {
```

[Table of Contents] [Description]

Path 444:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
      ...
259. try {
```

[Table of Contents] [Description]

Path 445:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
      ...
270. try {
```

[Table of Contents] [Description]

Path 446:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
      ...
281. try {
```

[Table of Contents] [Description]

Path 447:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
```

```
context) //posttypeimpl.java
    ...
292. try {
```

[Table of Contents] [Description]

Path 448:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
303. try {
```

[Table of Contents] [Description]

Path 449:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
314. try {
```

[Table of Contents] [Description]

Path 450:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //posttypeimpl.java
    ...
325. try {
```

[Table of Contents] [Description]

Path 451:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
950. public void handleText(final java.lang.String value) //posttypeimpl.java
    ...
956. try {
```

[Table of Contents] [Description]

Path 452:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1073. private void eatText1(final java.lang.String value) //posttypeimpl.java
      ...
1076. try {
```

[Table of Contents] [Description]

Path 453:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1083. private void eatText2(final java.lang.String value) //posttypeimpl.java
      ...
1086. try {
```

[Table of Contents] [Description]

Path 454:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1093. private void eatText3(final java.lang.String value) //posttypeimpl.java
      ...
1096. try {
```

[Table of Contents] [Description]

Path 455:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1103. private void eatText4(final java.lang.String value) //posttypeimpl.java
      ...
1106. try {
```

[Table of Contents] [Description]

Path 456:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1114. private void eatText5(final java.lang.String value) //posttypeimpl.java
      ...
1117. try {
```

[Table of Contents] [Description]

Path 457:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1124. private void eatText6(final java.lang.String value) //posttypeimpl.java
      ...
1127. try {
```

[Table of Contents] [Description]

Path 458:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1134. private void eatText7(final java.lang.String value) //posttypeimpl.java
      ...
1137. try {
```

[Table of Contents] [Description]

Path 459:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1145. private void eatText8(final java.lang.String value) //posttypeimpl.java
      ...
1148. try {
```

[Table of Contents] [Description]

Path 460:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1155. private void eatText9(final java.lang.String value) //posttypeimpl.java
      ...
1158. try {
```

[Table of Contents] [Description]

Path 461:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1165. private void eatText10(final java.lang.String value) //posttypeimpl.java
      ...
1168. try {
```

[Table of Contents] [Description]

Path 462:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1176. private void eatText11(final java.lang.String value) //posttypeimpl.java
      ...
1179. try {
```

[Table of Contents] [Description]

Path 463:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1186. private void eatText12(final java.lang.String value) //posttypeimpl.java
      ...
1189. try {
```

[Table of Contents] [Description]

Path 464:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1197. private void eatText13(final java.lang.String value) //posttypeimpl.java
      ...
1200. try {
```

[Table of Contents] [Description]

Path 465:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1208. private void eatText14(final java.lang.String value) //posttypeimpl.java
      ...
1211. try {
```

[Table of Contents] [Description]

Path 466:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
      ...
96. try {
```

[Table of Contents] [Description]

Path 467:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
      ...
106. try {
```

[Table of Contents] [Description]

Path 468:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
      ...
116. try {
```

[Table of Contents] [Description]

Path 469:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
      ...
126. try {
```

[Table of Contents] [Description]

Path 470:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
      ...
137. try {
```

[Table of Contents] [Description]

Path 471:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //ranktypeimpl.java
```

```
...
148. try {
```

[Table of Contents] [Description]

Path 472:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
459. public void handleText(final java.lang.String value) //ranktypeimpl.java
...
465. try {
```

[Table of Contents] [Description]

Path 473:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
514. private void eatText1(final java.lang.String value) //ranktypeimpl.java
...
517. try {
```

[Table of Contents] [Description]

Path 474:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
525. private void eatText2(final java.lang.String value) //ranktypeimpl.java
...
528. try {
```

[Table of Contents] [Description]

Path 475:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
536. private void eatText3(final java.lang.String value) //ranktypeimpl.java
...
539. try {
```

[Table of Contents] [Description]

Path 476:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
547. private void eatText4(final java.lang.String value) //ranktypeimpl.java
```

```
...
550. try {
```

[Table of Contents] [Description]

Path 477:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
558. private void eatText5(final java.lang.String value) //ranktypeimpl.java
...
561. try {
```

[Table of Contents] [Description]

Path 478:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
568. private void eatText6(final java.lang.String value) //ranktypeimpl.java
...
571. try {
```

[Table of Contents] [Description]

Path 479:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
251. public void handleText(final java.lang.String value) //threadlistimpl.java
...
257. try {
```

[Table of Contents] [Description]

Path 480:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
...
221. try {
```

[Table of Contents] [Description]

Path 481:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
232. try {
```

[Table of Contents] [Description]

Path 482:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
242. try {
```

[Table of Contents] [Description]

Path 483:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
251. try {
```

[Table of Contents] [Description]

Path 484:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
261. try {
```

[Table of Contents] [Description]

Path 485:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
272. try {
```

[Table of Contents] [Description]

Path 486:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
     ...
283. try {
```

[Table of Contents] [Description]

Path 487:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
     ...
294. try {
```

[Table of Contents] [Description]

Path 488:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
     ...
305. try {
```

[Table of Contents] [Description]

Path 489:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
     ...
316. try {
```

[Table of Contents] [Description]

Path 490:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
```

```
context) //threadtypeimpl.java
    ...
327. try {
```

[Table of Contents] [Description]

Path 491:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
338. try {
```

[Table of Contents] [Description]

Path 492:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
349. try {
```

[Table of Contents] [Description]

Path 493:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
360. try {
```

[Table of Contents] [Description]

Path 494:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
    ...
371. try {
```

[Table of Contents] [Description]

Path 495:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadtypeimpl.java
      ...
382. try {
```

[Table of Contents] [Description]

Path 496:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1113. public void handleText(final java.lang.String value) //threadtypeimpl.java
      ...
1119. try {
```

[Table of Contents] [Description]

Path 497:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1256. private void eatText1(final java.lang.String value) //threadtypeimpl.java
      ...
1259. try {
```

[Table of Contents] [Description]

Path 498:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1266. private void eatText2(final java.lang.String value) //threadtypeimpl.java
      ...
1269. try {
```

[Table of Contents] [Description]

Path 499:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1277. private void eatText3(final java.lang.String value) //threadtypeimpl.java
      ...
1280. try {
```

[Table of Contents] [Description]

Path 500:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1287. private void eatText4(final java.lang.String value) //threadtypeimpl.java
      ...
1290. try {
```

[Table of Contents] [Description]

Path 501:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1298. private void eatText5(final java.lang.String value) //threadtypeimpl.java
      ...
1301. try {
```

[Table of Contents] [Description]

Path 502:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1309. private void eatText6(final java.lang.String value) //threadtypeimpl.java
      ...
1312. try {
```

[Table of Contents] [Description]

Path 503:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1320. private void eatText7(final java.lang.String value) //threadtypeimpl.java
      ...
1323. try {
```

[Table of Contents] [Description]

Path 504:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1330. private void eatText8(final java.lang.String value) //threadtypeimpl.java
      ...
1333. try {
```

[Table of Contents] [Description]

Path 505:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1341. private void eatText9(final java.lang.String value) //threadtypeimpl.java
      ...
1344. try {
```

[Table of Contents] [Description]

Path 506:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1351. private void eatText10(final java.lang.String value) //threadtypeimpl.java
      ...
1354. try {
```

[Table of Contents] [Description]

Path 507:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1362. private void eatText11(final java.lang.String value) //threadtypeimpl.java
      ...
1365. try {
```

[Table of Contents] [Description]

Path 508:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1373. private void eatText12(final java.lang.String value) //threadtypeimpl.java
      ...
1376. try {
```

[Table of Contents] [Description]

Path 509:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1383. private void eatText13(final java.lang.String value) //threadtypeimpl.java
      ...
1386. try {
```

[Table of Contents] [Description]

Path 510:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1393. private void eatText14(final java.lang.String value) //threadtypeimpl.java
      ...
1396. try {
```

[Table of Contents] [Description]

Path 511:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1404. private void eatText15(final java.lang.String value) //threadtypeimpl.java
      ...
1407. try {
```

[Table of Contents] [Description]

Path 512:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1415. private void eatText16(final java.lang.String value) //threadtypeimpl.java
      ...
1418. try {
```

[Table of Contents] [Description]

Path 513:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
278. public void handleText(final java.lang.String value) //threadwatchlistimpl.java
     ...
284. try {
```

[Table of Contents] [Description]

Path 514:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
101. try {
```

[Table of Contents] [Description]

Path 515:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
112. try {
```

[Table of Contents] [Description]

Path 516:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
123. try {
```

[Table of Contents] [Description]

Path 517:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
134. try {
```

[Table of Contents] [Description]

Path 518:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
145. try {
```

[Table of Contents] [Description]

Path 519:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
156. try {
```

[Table of Contents] [Description]

Path 520:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void serializeBody(com.mvnforum.jaxb.db.impl.runtime.XMLSerializer
context) //threadwatchtypeimpl.java
    ...
167. try {
```

[Table of Contents] [Description]

Path 521:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
519. public void handleText(final java.lang.String value) //threadwatchtypeimpl.java
    ...
525. try {
```

[Table of Contents] [Description]

Path 522:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
587. private void eatText1(final java.lang.String value) //threadwatchtypeimpl.java
    ...
590. try {
```

[Table of Contents] [Description]

Path 523:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
598. private void eatText2(final java.lang.String value) //threadwatchtypeimpl.java
    ...
601. try {
```

[Table of Contents] [Description]

Path 524:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
608. private void eatText3(final java.lang.String value) //threadwatchtypeimpl.java
     ...
611. try {
```

[Table of Contents] [Description]

Path 525:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
619. private void eatText4(final java.lang.String value) //threadwatchtypeimpl.java
     ...
622. try {
```

[Table of Contents] [Description]

Path 526:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. private void eatText5(final java.lang.String value) //threadwatchtypeimpl.java
     ...
633. try {
```

[Table of Contents] [Description]

Path 527:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
640. private void eatText6(final java.lang.String value) //threadwatchtypeimpl.java
     ...
643. try {
```

[Table of Contents] [Description]

Path 528:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
650. private void eatText7(final java.lang.String value) //threadwatchtypeimpl.java
     ...
653. try {
```

[Table of Contents] [Description]

Path 529:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
196. private UnmarshallingEventHandler spawnChild( Class clazz, int
memento ) //abstractunmarshallingeventhandlerimpl.java
     ...
199. try {
```

[Table of Contents] [Description]

Path 530:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
199. private static String getVersion(JAXBContext c) throws
JAXBException //grammarinfofacade.java
     200. try {
```

[Table of Contents] [Description]

Path 531:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public final UnmarshallingEventHandler createUnmarshaller
( //grammarinfoimpl.java
     84. if(intfCls==null) return null;
```

[Table of Contents] [Description]

Path 532:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
117. public final Class getDefaultImplementation( Class
javaContentInterface ) //grammarinfoimpl.java
     118. try {
```

[Table of Contents] [Description]

Path 533:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
     ...
100. try {
```

[Table of Contents] [Description]

Path 534:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
      134. try {
```

[Table of Contents] [Description]

Path 535:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
153. private void write( XMLSerializable obj, ContentHandler
writer ) //marshallerimpl.java
      ...
      156. try {
```

[Table of Contents] [Description]

Path 536:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
187. protected CharacterEscapeHandler createEscapeHandler( String
encoding ) //marshallerimpl.java
      ...
      198. try {
```

[Table of Contents] [Description]

Path 537:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
236. public XMLWriter createWriter( OutputStream os, String encoding ) throws
JAXBException //marshallerimpl.java
      237. try {
```

[Table of Contents] [Description]

Path 538:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. public static void validate( DefaultJAXBContextImpl jaxbCtx, ValidationContext
context, ValidatableObject vo ) //msvvalidator.java
      ...
      76. try {
```

[Table of Contents] [Description]

Path 539:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
425. public void reportError( ValidationEvent ve ) throws
AbortSerializationException //saxmarshaller.java
      ...
428. try {
```

[Table of Contents] [Description]

Path 540:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
564. public void handleEvent(ValidationEvent event, boolean canRecover ) throws
SAXException //saxunmarshallerhandlerimpl.java
      ...
566. try {
```

[Table of Contents] [Description]

Path 541:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. private SAXUnmarshallerHandler createUnmarshallerHandler( Locator
locator ) //unmarshallerimpl.java
      ...
99. try {
```

[Table of Contents] [Description]

Path 542:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
117. protected Object unmarshal( XMLReader reader, InputSource source ) throws
JAXBException //unmarshallerimpl.java
      ...
139. try {
```

[Table of Contents] [Description]

Path 543:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
158. public final Object unmarshal( Node node ) throws
JAXBException //unmarshallerimpl.java
```

```
159. try {
```

[Table of Contents] [Description]

Path 544:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
36. public UnmarshallingEventHandlerAdaptor(UnmarshallingContext
_ctxt,ContentHandler _handler) throws
SAXException //unmarshallingeventhandleradaptor.java
    ...
41. try {
```

[Table of Contents] [Description]

Path 545:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
61. public void enterElement(String uri, String local, String qname, Attributes
atts) throws SAXException //unmarshallingeventhandleradaptor.java
    ...
64. try {
```

[Table of Contents] [Description]

Path 546:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public void leaveElement(String uri, String local, String qname) throws
SAXException //unmarshallingeventhandleradaptor.java
    76. try {
```

[Table of Contents] [Description]

Path 547:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public void leaveElement(String uri, String local, String qname) throws
SAXException //unmarshallingeventhandleradaptor.java
    ...
87. try {
```

[Table of Contents] [Description]

Path 548:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. public void text(String s) throws
SAXException //unmarshallingeventhandleradaptor.java
     |110. try {
```

[Table of Contents] [Description]

Path 549:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
186. public void reportEvent( ValidatableObject source, ValidationEvent event )
throws AbortSerializationException //validationcontext.java
     |...
     |189. try {
```

[Table of Contents] [Description]

Path 550:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
68. public boolean handleEvent( ValidationEvent e ) //validatorimpl.java
     |...
     |73. try {
```

[Table of Contents] [Description]

Path 551:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. private boolean validate( Object o, boolean validateId ) //validatorimpl.java
     |...
     |110. try {
```

[Table of Contents] [Description]

Path 552:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
14. public class XMLUtil //xmlutil.java
     |...
     |24. try {
```

[Table of Contents] [Description]

Path 553:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public static void main(String[] args) //md5.java
       ...
110. try {
```

[Table of Contents] [Description]

Path 554:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public static void main(String[] args) //md5.java
       ...
117. try {
```

[Table of Contents] [Description]

Path 555:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
159. public static synchronized String getMD5_Base64(String input) //md5.java
       ...
174. try {
```

[Table of Contents] [Description]

Path 556:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
160. try {
```

[Table of Contents] [Description]

Path 557:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
       ...
190. try
```

[Table of Contents] [Description]

Path 558:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
264. public static ArrayList getContent(String fileName) //converturl.java
```

```
...
270. try {
```

[Table of Contents] [Description]

Path 559:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
297. public static void ExportContentToFile(ArrayList lines, String
fileName) //converturl.java
...
299. try {
```

[Table of Contents] [Description]

Path 560:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
219. private void processBundle(final String bundleFile, //localizingkeyfinder.java
...
224. try {
```

[Table of Contents] [Description]

Path 561:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
240. private void loadResourceMap(FileInputStream ins) throws
BuildException //localizingkeyfinder.java
241. try {
```

[Table of Contents] [Description]

Path 562:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
302. private void translate() throws BuildException //localizingkeyfinder.java
...
316. try {
```

[Table of Contents] [Description]

Path 563:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
424. private void processBundle(final String bundleFile, final int
i, //translate.java
      ...
428. try {
```

[Table of Contents] [Description]

Path 564:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
449. private void loadResourceMap(FileInputStream ins) throws
BuildException //translate.java
      450. try {
```

[Table of Contents] [Description]

Path 565:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
514. private void translate() throws BuildException //translate.java
      ...
526. try {
```

[Table of Contents] [Description]

Path 566:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
424. private void processBundle(final String bundleFile, final int
i, //translatetojstl.java
      ...
428. try {
```

[Table of Contents] [Description]

Path 567:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
449. private void loadResourceMap(FileInputStream ins) throws
BuildException //translatetojstl.java
      450. try {
```

[Table of Contents] [Description]

Path 568:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
514. private void translate() throws BuildException //translatetojstl.java
      ...
526. try {
```

[Table of Contents] [Description]

Path 569:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
200. private static void setMVNForumHome(String home) //mvnforumconfig.java
      ...
203. try {
```

[Table of Contents] [Description]

Path 570:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public void contextDestroyed(ServletContextEvent
event) //mvnforumcontextlistener.java
      ...
100. try {
```

[Table of Contents] [Description]

Path 571:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public static boolean canViewAnyForumInCategory(int categoryID,
MVNForumPermission permission) //myutil.java
215. try {
```

[Table of Contents] [Description]

Path 572:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
351. public static void ensureCorrectCurrentPassword(GenericRequest request, String
password, String passwordMD5) //myutil.java
      ...
365. try {
```

[Table of Contents] [Description]

Path 573:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
423. public static void writeMvnForumImage(HttpServletResponse response) throws
IOException //myutil.java
     ...
427. try {
```

[Table of Contents] [Description]

Path 574:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
423. public static void writeMvnForumImage(HttpServletResponse response) throws
IOException //myutil.java
     ...
439. try {
```

[Table of Contents] [Description]

Path 575:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
446. public static void writeMvnCoreImage(HttpServletResponse response) throws
IOException //myutil.java
     ...
450. try {
```

[Table of Contents] [Description]

Path 576:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
446. public static void writeMvnCoreImage(HttpServletResponse response) throws
IOException //myutil.java
     ...
463. try {
```

[Table of Contents] [Description]

Path 577:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
470. public static void checkClassName(Locale locale, String className, boolean
required) throws BadInputException //myutil.java
     ...
475. try {
```

[Table of Contents] [Description]

Path 578:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
502. public static Hashtable checkMembers(String[] memberNames, Locale
locale) //myutil.java
     ...
513. try {
```

[Table of Contents] [Description]

Path 579:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public String preLogin(HttpServletRequest request, HttpServletResponse
response) //requestprocessordefault.java
     68. try {
```

[Table of Contents] [Description]

Path 580:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
128. public String process(HttpServletRequest request, HttpServletResponse
response) //adminmoduleprocessor.java
     ...
144. try {
```

[Table of Contents] [Description]

Path 581:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
226. public void addAttachment(String strAttachmentID, //attachmentxml.java
     ...
264. try {
```

[Table of Contents] [Description]

Path 582:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
314. public static void exportAttachmentList(XMLWriter xmlWriter, int
parentPostID) //attachmentxml.java
```

```
...
327. try {
```

[Table of Contents] [Description]

Path 583:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
113. public void addCategory(String categoryName, //categoryxml.java
...
130. try {
```

[Table of Contents] [Description]

Path 584:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
113. public void addCategory(String categoryName, //categoryxml.java
...
152. try {
```

[Table of Contents] [Description]

Path 585:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
194. public void addCategoryWatch(String memberName, //categoryxml.java
...
212. try {
```

[Table of Contents] [Description]

Path 586:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
239. public static void exportCategoryWatchesForCategory(XMLWriter xmlWriter, int
categoryID) //categoryxml.java
...
252. try {
```

[Table of Contents] [Description]

Path 587:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. public static void exportCategory(XMLWriter xmlWriter, int
categoryID) //categoryxml.java
     ...
300. try {
```

[Table of Contents] [Description]

Path 588:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public static void exportSubCategoryList(XMLWriter xmlWriter, int
parentCategoryID) //categoryxml.java
     ...
352. try {
```

[Table of Contents] [Description]

Path 589:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public static void exportSubCategoryList(XMLWriter xmlWriter, int
parentCategoryID) //categoryxml.java
     ...
357. try {
```

[Table of Contents] [Description]

Path 590:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
     ...
258. try {
```

[Table of Contents] [Description]

Path 591:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
     ...
279. try {
```

[Table of Contents] [Description]

Path 592:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1252. public void processEditTemplate(GenericRequest
request) //configurationwebhandler.java
      ...
1297. try {
```

[Table of Contents] [Description]

Path 593:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
91. public static void exportXmlZip(HttpServletRequest
request) //exportwebhandler.java
      ...
107. try {
```

[Table of Contents] [Description]

Path 594:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
157. try {
```

[Table of Contents] [Description]

Path 595:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
173. try {
```

[Table of Contents] [Description]

Path 596:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
```

```
HttpServletResponse response) //exportwebhandler.java
    ...
180. try {
```

[Table of Contents] [Description]

Path 597:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
241. private static void sendToUser(BufferedInputStream
inputStream, //exportwebhandler.java
    ...
245. try {
```

[Table of Contents] [Description]

Path 598:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
241. private static void sendToUser(BufferedInputStream
inputStream, //exportwebhandler.java
    ...
252. try {
```

[Table of Contents] [Description]

Path 599:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
80. public static void exportXml(String filename, HttpServletRequest
request, //exportwebhelper.java
    ...
89. try {
```

[Table of Contents] [Description]

Path 600:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
80. public static void exportXml(String filename, HttpServletRequest
request, //exportwebhelper.java
    ...
98. try {
```

[Table of Contents] [Description]

Path 601:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
80. public static void exportXml(String filename, HttpServletRequest
request, //exportwebhelper.java
      ...
110. try {
```

[Table of Contents] [Description]

Path 602:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
      ...
133. try {
```

[Table of Contents] [Description]

Path 603:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
      ...
164. try {
```

[Table of Contents] [Description]

Path 604:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
249. private static void createBackupXml(XMLWriter xmlWriter, //exportwebhelper.java
      ...
253. try {
```

[Table of Contents] [Description]

Path 605:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public static Collection execSqlQuery(String query) //exportwebhelper.java
      ...
290. try {
```

[Table of Contents] [Description]

Path 606:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
128. public void addFavoriteThread(String memberName, //favoritethreadxml.java
     ...
144. try {
```

[Table of Contents] [Description]

Path 607:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
103. public void process(HttpServletRequest request, HttpServletResponse
response) //forumadminservlet.java
     ...
111. try {
```

[Table of Contents] [Description]

Path 608:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
103. public void process(HttpServletRequest request, HttpServletResponse
response) //forumadminservlet.java
     ...
115. try {
```

[Table of Contents] [Description]

Path 609:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
     ...
130. try {
```

[Table of Contents] [Description]

Path 610:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
163. public void prepareDelete(GenericRequest request) //forumwebhandler.java
```

```
...
175. try {
```

[Table of Contents] [Description]

Path 611:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void prepareEdit(GenericRequest request, GenericResponse
response) //forumwebhandler.java
...
257. try {
```

[Table of Contents] [Description]

Path 612:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
273. public void processUpdate(GenericRequest request, GenericResponse
response) //forumwebhandler.java
...
320. try {
```

[Table of Contents] [Description]

Path 613:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void addForum(String lastPostMemberName, String
forumName, //forumxml.java
...
149. try {
```

[Table of Contents] [Description]

Path 614:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
121. public void addForum(String lastPostMemberName, String
forumName, //forumxml.java
...
183. try {
```

[Table of Contents] [Description]

Path 615:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
218. public void addMemberForumPermission(String memberName, String
permission) //forumxml.java
      ...
230. try {
```

[Table of Contents] [Description]

Path 616:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
218. public void addMemberForumPermission(String memberName, String
permission) //forumxml.java
      ...
236. try {
```

[Table of Contents] [Description]

Path 617:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
259. public void addGroupForumPermission(String groupName, String
permission) //forumxml.java
      ...
271. try {
```

[Table of Contents] [Description]

Path 618:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
259. public void addGroupForumPermission(String groupName, String
permission) //forumxml.java
      ...
277. try {
```

[Table of Contents] [Description]

Path 619:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public void addGuestMemberForumPermission(String permission) //forumxml.java
      ...
291. try {
```

[Table of Contents] [Description]

Path 620:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public void addGuestMemberForumPermission(String permission) //forumxml.java
      ...
296. try {
```

[Table of Contents] [Description]

Path 621:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
304. public void addRegisteredMembersGroupForumPermission(String
permission) //forumxml.java
      ...
311. try {
```

[Table of Contents] [Description]

Path 622:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
304. public void addRegisteredMembersGroupForumPermission(String
permission) //forumxml.java
      ...
316. try {
```

[Table of Contents] [Description]

Path 623:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
346. public void addForumWatch(String memberName, //forumxml.java
      ...
364. try {
```

[Table of Contents] [Description]

Path 624:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
425. public static void exportForumWatchesForForum(XMLWriter xmlWriter, int
forumID) //forumxml.java
```

```
...
437. try {
```

[Table of Contents] [Description]

Path 625:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
474. public static void exportMemberForumPermissionsForForum(XMLWriter xmlWriter,
int forumID) //forumxml.java
...
485. try {
```

[Table of Contents] [Description]

Path 626:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
507. public static void exportGroupForumPermissionsForForum(XMLWriter xmlWriter, int
forumID) //forumxml.java
...
518. try {
```

[Table of Contents] [Description]

Path 627:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
539. public static void exportForum(XMLWriter xmlWriter, int
forumID) //forumxml.java
...
553. try {
```

[Table of Contents] [Description]

Path 628:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
625. public static void exportForumList(XMLWriter xmlWriter, int
parentCategoryID) //forumxml.java
...
635. try {
```

[Table of Contents] [Description]

Path 629:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
625. public static void exportForumList(XMLWriter xmlWriter, int
parentCategoryID) //forumxml.java
     ...
640. try {
```

[Table of Contents] [Description]

Path 630:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
412. public void sendMail(GenericRequest request) //generaladmintaskswebhandler.java
     ...
515. try {
```

[Table of Contents] [Description]

Path 631:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
525. public void sendActivateMailToAll(GenericRequest
request) //generaladmintaskswebhandler.java
     ...
543. try {
```

[Table of Contents] [Description]

Path 632:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
556. private String processMailTemplate(MemberBean memberBean, String
message) //generaladmintaskswebhandler.java
     557. try {
```

[Table of Contents] [Description]

Path 633:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
     ...
778. try {
```

[Table of Contents] [Description]

Path 634:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
      |
      ...
      799. try {
```

[Table of Contents] [Description]

Path 635:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
78. public void prepareList(GenericRequest request) //groupforumwebhandler.java
      |
      ...
      91. try {
```

[Table of Contents] [Description]

Path 636:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
186. public void prepareAssignGroupToForum(GenericRequest
request) //groupforumwebhandler.java
      |
      ...
      198. try {
```

[Table of Contents] [Description]

Path 637:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
186. public void prepareAssignGroupToForum(GenericRequest
request) //groupforumwebhandler.java
      |
      ...
      213. try {
```

[Table of Contents] [Description]

Path 638:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
      |
```

```
...
91. try {
```

[Table of Contents] [Description]

Path 639:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
189. public void processUpdateGroupOwner(GenericRequest
request) //groupswebhandler.java
...
229. try {
```

[Table of Contents] [Description]

Path 640:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
...
141. try {
```

[Table of Contents] [Description]

Path 641:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
...
153. try {
```

[Table of Contents] [Description]

Path 642:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
...
169. try {
```

[Table of Contents] [Description]

Path 643:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
207. public void addGroupPermission(String permission) //groupxml.java
      ...
213. try {
```

[Table of Contents] [Description]

Path 644:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
207. public void addGroupPermission(String permission) //groupxml.java
      ...
218. try {
```

[Table of Contents] [Description]

Path 645:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
225. public static void addRegisteredMembersGroupPermission(String
permission) //groupxml.java
      ...
229. try {
```

[Table of Contents] [Description]

Path 646:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public static void addGroupPermission(String groupname, String
permission) //groupxml.java
      ...
241. try {
```

[Table of Contents] [Description]

Path 647:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
267. public void addMemberGroup(String memberName, String privilege, //groupxml.java
      ...
280. try {
```

[Table of Contents] [Description]

Path 648:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
295. public static void exportGlobalPermissionsForGroup(XMLWriter xmlWriter, int
groupID) //groupxml.java
     ...
305. try {
```

[Table of Contents] [Description]

Path 649:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
321. public static void exportGroupMembersForGroup(XMLWriter xmlWriter, int
groupID) //groupxml.java
     ...
332. try {
```

[Table of Contents] [Description]

Path 650:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
359. public static void exportGroup(XMLWriter xmlWriter, int
groupID) //groupxml.java
     ...
369. try {
```

[Table of Contents] [Description]

Path 651:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
410. public static void exportGroupList(XMLWriter xmlWriter) //groupxml.java
     ...
425. try {
```

[Table of Contents] [Description]

Path 652:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
410. public static void exportGroupList(XMLWriter xmlWriter) //groupxml.java
     ...
430. try {
```

[Table of Contents] [Description]

Path 653:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
145. try {
```

[Table of Contents] [Description]

Path 654:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
200. try {
```

[Table of Contents] [Description]

Path 655:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
208. try {
```

[Table of Contents] [Description]

Path 656:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
217. try {
```

[Table of Contents] [Description]

Path 657:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
```

```
...
304. try {
```

[Table of Contents] [Description]

Path 658:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
434. public static void addFinalErrorHandling(HttpServletRequest request, boolean
resetAllData) //importwebhandler.java
...
452. try {
```

[Table of Contents] [Description]

Path 659:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
82. protected static int execUpdateQuery(String query) throws
DatabaseException //importwebhelper.java
...
87. try {
```

[Table of Contents] [Description]

Path 660:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public void prepareAssignForumToMember(GenericRequest request, GenericResponse
response) //memberforumwebhandler.java
...
88. try {
```

[Table of Contents] [Description]

Path 661:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
108. public void prepareAssignMemberToForum(GenericRequest
request) //memberforumwebhandler.java
...
121. try {
```

[Table of Contents] [Description]

Path 662:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
108. public void prepareAssignMemberToForum(GenericRequest
request) //memberforumwebhandler.java
      ...
140. try {
```

[Table of Contents] [Description]

Path 663:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
271. public void prepareList(GenericRequest request) //memberforumwebhandler.java
      ...
282. try {
```

[Table of Contents] [Description]

Path 664:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
271. public void prepareList(GenericRequest request) //memberforumwebhandler.java
      ...
296. try {
```

[Table of Contents] [Description]

Path 665:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
271. public void prepareList(GenericRequest request) //memberforumwebhandler.java
      ...
303. try {
```

[Table of Contents] [Description]

Path 666:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void processAdd(GenericRequest request) //membergroupwebhandler.java
      ...
115. try {
```

[Table of Contents] [Description]

Path 667:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
103. public void processAdd(GenericRequest request) //memberwebhandler.java
      ...
213. try {
```

[Table of Contents] [Description]

Path 668:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
240. public void processUpdateMemberStatus(GenericRequest
request) //memberwebhandler.java
      ...
266. try {
```

[Table of Contents] [Description]

Path 669:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
274. public void prepareEdit(GenericRequest request) //memberwebhandler.java
      ...
286. try {
```

[Table of Contents] [Description]

Path 670:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. public void processEdit(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
342. try {
```

[Table of Contents] [Description]

Path 671:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
      ...
388. try {
```

[Table of Contents] [Description]

Path 672:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
     ...
397. try {
```

[Table of Contents] [Description]

Path 673:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
     ...
404. try {
```

[Table of Contents] [Description]

Path 674:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
     ...
418. try {
```

[Table of Contents] [Description]

Path 675:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
     ...
452. try {
```

[Table of Contents] [Description]

Path 676:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
475. public void processUpdateMemberTitle(GenericRequest request, GenericResponse
response) //memberwebhandler.java
     ...
492. try {
```

[Table of Contents] [Description]

Path 677:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
500. public void processResetMemberSignature(GenericRequest
request) //memberwebhandler.java
      |...
      |509. try {
```

[Table of Contents] [Description]

Path 678:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
519. public void processResetMemberAvatar(GenericRequest
request) //memberwebhandler.java
      |...
      |528. try {
```

[Table of Contents] [Description]

Path 679:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
538. public void processResetMemberActivation(GenericRequest
request) //memberwebhandler.java
      |...
      |547. try {
```

[Table of Contents] [Description]

Path 680:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
557. public void prepareDeleteMember(GenericRequest request) //memberwebhandler.java
      |...
      |578. try {
```

[Table of Contents] [Description]

Path 681:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
588. public void processDeleteMember(GenericRequest request) //memberwebhandler.java
```

```
...
615. try {
```

[Table of Contents] [Description]

Path 682:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
...
649. try {
```

[Table of Contents] [Description]

Path 683:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
...
718. try {
```

[Table of Contents] [Description]

Path 684:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
738. public void prepareListPendingUsers(GenericRequest
request) //memberwebhandler.java
...
756. try {
```

[Table of Contents] [Description]

Path 685:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
781. public void processListPendingUsers(GenericRequest
request) //memberwebhandler.java
...
800. try {
```

[Table of Contents] [Description]

Path 686:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
853. try {
```

[Table of Contents] [Description]

Path 687:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
871. try {
```

[Table of Contents] [Description]

Path 688:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
883. try {
```

[Table of Contents] [Description]

Path 689:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1042. public void prepareViewPermission(GenericRequest
request) //memberwebhandler.java
      ...
1054. try {
```

[Table of Contents] [Description]

Path 690:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1075. public void prepareChangePassword(GenericRequest
request ) //memberwebhandler.java
      ...
```

```
1095. try {
```

[Table of Contents] [Description]

Path 691:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1105. public void processChangePassword(GenericRequest
request ) //memberwebhandler.java
      ...
1145. try {
```

[Table of Contents] [Description]

Path 692:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
      ...
310. try {
```

[Table of Contents] [Description]

Path 693:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
      ...
362. try {
```

[Table of Contents] [Description]

Path 694:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
462. public void addMemberPermission(String permission) //memberxml.java
      ...
468. try {
```

[Table of Contents] [Description]

Path 695:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
462. public void addMemberPermission(String permission) //memberxml.java
```

```
...
473. try {
```

[Table of Contents] [Description]

Path 696:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
480. public static void addGuestMemberPermission(String permission) //memberxml.java
...
483. try {
```

[Table of Contents] [Description]

Path 697:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
480. public static void addGuestMemberPermission(String permission) //memberxml.java
...
488. try {
```

[Table of Contents] [Description]

Path 698:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
495. public static void addAdminMemberPermission(String permission) //memberxml.java
...
498. try {
```

[Table of Contents] [Description]

Path 699:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
495. public static void addAdminMemberPermission(String permission) //memberxml.java
...
503. try {
```

[Table of Contents] [Description]

Path 700:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
510. public static void addMemberPermission(String memberName, String
```

```
permission) //memberxml.java
    ...
514. try {
```

[Table of Contents] [Description]

Path 701:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
510. public static void addMemberPermission(String memberName, String
permission) //memberxml.java
    ...
519. try {
```

[Table of Contents] [Description]

Path 702:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
543. public void addMessageFolder(String folderName, String
folderOrder, //memberxml.java
    ...
556. try {
```

[Table of Contents] [Description]

Path 703:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
593. public void addGlobalWatch(String watchType, String
watchOption, //memberxml.java
    ...
608. try {
```

[Table of Contents] [Description]

Path 704:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. public static void exportMessageFoldersForMember(XMLWriter xmlWriter, String
memberName) //memberxml.java
    ...
632. try {
```

[Table of Contents] [Description]

Path 705:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
640. public static void exportMessageFoldersForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
     |...
     |650. try {
```

[Table of Contents] [Description]

Path 706:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
677. public static void exportGlobalPermissionsForMember(XMLWriter xmlWriter, String
memberName) //memberxml.java
     |...
     |679. try {
```

[Table of Contents] [Description]

Path 707:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
687. public static void exportGlobalPermissionsForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
     |...
     |697. try {
```

[Table of Contents] [Description]

Path 708:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
713. public static void exportGlobalWatchesForMember(XMLWriter xmlWriter, String
memberName) //memberxml.java
     |...
     |715. try {
```

[Table of Contents] [Description]

Path 709:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
723. public static void exportGlobalWatchesForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
     |
```

```
...
734. try {
```

[Table of Contents] [Description]

Path 710:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
767. public static void exportMember(XMLWriter xmlWriter, String
memberName) //memberxml.java
...
769. try {
```

[Table of Contents] [Description]

Path 711:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
777. public static void exportMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
...
798. try {
```

[Table of Contents] [Description]

Path 712:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
...
1002. try {
```

[Table of Contents] [Description]

Path 713:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
...
1007. try {
```

[Table of Contents] [Description]

Path 714:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
      ...
403. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 715:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
273. public static void exportPost(XMLWriter xmlWriter, int postID) //postxml.java
      ...
286. try {
```

[Table of Contents] [Description]

Path 716:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
351. public static void exportPostList_FirstPosts(XMLWriter xmlWriter, int
parentThreadID) //postxml.java
      ...
362. try {
```

[Table of Contents] [Description]

Path 717:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
351. public static void exportPostList_FirstPosts(XMLWriter xmlWriter, int
parentThreadID) //postxml.java
      ...
367. try {
```

[Table of Contents] [Description]

Path 718:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
381. public static void exportPostList_Replies(XMLWriter xmlWriter, int
parentPostID) //postxml.java
      ...
391. try {
```

[Table of Contents] [Description]

Path 719:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
381. public static void exportPostList_Replies(XMLWriter xmlWriter, int
parentPostID) //postxml.java
      ...
396. try {
```

[Table of Contents] [Description]

Path 720:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
95. public void addRank(String rankMinPosts, String rankLevel, //rankxml.java
      ...
109. try {
```

[Table of Contents] [Description]

Path 721:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
133. public static void exportRankList(XMLWriter xmlWriter) //rankxml.java
      ...
143. try {
```

[Table of Contents] [Description]

Path 722:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. public void addThread(String memberName, String
lastPostMemberName, //threadxml.java
      ...
176. try {
```

[Table of Contents] [Description]

Path 723:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
232. public void addThreadWatch(String memberName, //threadxml.java
      ...
250. try {
```

[Table of Contents] [Description]

Path 724:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
273. public void addFavoriteThread(String memberName, //threadxml.java
     ...
290. try {
```

[Table of Contents] [Description]

Path 725:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
339. public static void exportThreadWatchesForThread(XMLWriter xmlWriter, int
threadID) //threadxml.java
     ...
350. try {
```

[Table of Contents] [Description]

Path 726:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
387. public static void exportFavoriteThreadsForThread(XMLWriter xmlWriter, int
threadID) //threadxml.java
     ...
399. try {
```

[Table of Contents] [Description]

Path 727:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
430. public static void exportThread(XMLWriter xmlWriter, int
threadID) //threadxml.java
     ...
444. try {
```

[Table of Contents] [Description]

Path 728:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
523. public static void exportThreadList(XMLWriter xmlWriter, int
```

```
parentForumID) //threadxml.java
      ...
533. try {
```

[Table of Contents] [Description]

Path 729:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
523. public static void exportThreadList(XMLWriter xmlWriter, int
parentForumID) //threadxml.java
      ...
538. try {
```

[Table of Contents] [Description]

Path 730:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
118. public static java.sql.Date stringToSqlDate(String s) //xmlutil.java
      ...
132. try {
```

[Table of Contents] [Description]

Path 731:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
185. public static java.sql.Timestamp stringToSqlTimestamp(String s) //xmlutil.java
      ...
200. try {
```

[Table of Contents] [Description]

Path 732:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public XMLWriter(String indentString, File outputFile) //xmlwriter.java
      ...
104. try {
```

[Table of Contents] [Description]

Path 733:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
106. public static void importXml(File importFile, //importjive.java
     ...
125. try {
```

[Table of Contents] [Description]

Path 734:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     ...
140. try {
```

[Table of Contents] [Description]

Path 735:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
184. protected static void processXml(File inputFile, Calendar
importTime) //importjive.java
     ...
372. try {
```

[Table of Contents] [Description]

Path 736:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
184. protected static void processXml(File inputFile, Calendar
importTime) //importjive.java
     ...
374. try {
```

[Table of Contents] [Description]

Path 737:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
184. protected static void processXml(File inputFile, Calendar
importTime) //importjive.java
     ...
386. try {
```

[Table of Contents] [Description]

Path 738:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      |...
      |171. try {
```

[Table of Contents] [Description]

Path 739:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      |...
      |186. try {
```

[Table of Contents] [Description]

Path 740:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      |...
      |201. try {
```

[Table of Contents] [Description]

Path 741:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
216. public void addJiveForumGroup(String groupname, String
jivePermission) //jiveforumxml.java
      |...
      |228. try {
```

[Table of Contents] [Description]

Path 742:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
171. private void updateAdmin(String username, String mvnPassword, String
email, //jiveuserxml.java
      |
```

```
...
187. try {
```

[Table of Contents] [Description]

Path 743:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
...
224. try {
```

[Table of Contents] [Description]

Path 744:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
...
238. try {
```

[Table of Contents] [Description]

Path 745:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
...
252. try {
```

[Table of Contents] [Description]

Path 746:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
268. public void addJiveGroupPermission(String groupname, String
jivePermission) //jivexml.java
...
276. try {
```

[Table of Contents] [Description]

Path 747:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public static void importXml(File importFile, //importmvnforum.java
       ...
111. try {
```

[Table of Contents] [Description]

Path 748:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public static void importXml(File importFile, //importmvnforum.java
       ...
126. try {
```

[Table of Contents] [Description]

Path 749:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
       ...
182. try {
```

[Table of Contents] [Description]

Path 750:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
       ...
201. try {
```

[Table of Contents] [Description]

Path 751:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
       ...
216. try {
```

[Table of Contents] [Description]

Path 752:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
278. try {
```

[Table of Contents] [Description]

Path 753:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
289. try {
```

[Table of Contents] [Description]

Path 754:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
346. protected static void processXml(File inputFile, Calendar
importTime) //importmvnforum.java
     ...
652. try {
```

[Table of Contents] [Description]

Path 755:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
188. public static void checkGuestMember() throws CreateException,
DuplicateKeyException, //mvnforumxml.java
     ...
196. try {
```

[Table of Contents] [Description]

Path 756:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
63. static ArrayList getGroupPermissionsInStepWithChannel(int memberID) throws
DatabaseException //mvncmspermissionwebhelper.java
     ...
```

```
78. try {
```

[Table of Contents] [Description]

Path 757:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. static ArrayList getMemberPermissions(int
memberID) //mvnforumpermissionwebhelper.java
    ...
82. try {
```

[Table of Contents] [Description]

Path 758:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
102. static ArrayList getGroupPermissions(int
memberID) //mvnforumpermissionwebhelper.java
    ...
120. try {
```

[Table of Contents] [Description]

Path 759:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
140. static ArrayList getMemberPermissionsInForums(int
memberID) //mvnforumpermissionwebhelper.java
    ...
155. try {
```

[Table of Contents] [Description]

Path 760:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. static ArrayList getGroupPermissionsInForums(int
memberID) //mvnforumpermissionwebhelper.java
    ...
195. try {
```

[Table of Contents] [Description]

Path 761:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
217. static ArrayList getPermissionsForGroupGuest
() //mvnforumpermissionwebhelper.java
    ...
    228. try {
```

[Table of Contents] [Description]

Path 762:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
247. static ArrayList getPermissionsForGroupGuestInForums
() //mvnforumpermissionwebhelper.java
    ...
    263. try {
```

[Table of Contents] [Description]

Path 763:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    103. try {
```

[Table of Contents] [Description]

Path 764:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
198. public OnlineUser getAnonymousUser(GenericRequest
request) //onlineuserfactoryimpl.java
    ...
    209. try {
```

[Table of Contents] [Description]

Path 765:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
    ...
    271. try {
```

[Table of Contents] [Description]

Path 766:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
     ...
272. try {
```

[Table of Contents] [Description]

Path 767:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
     ...
279. try {
```

[Table of Contents] [Description]

Path 768:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
     ...
286. try {
```

[Table of Contents] [Description]

Path 769:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
     ...
293. try {
```

[Table of Contents] [Description]

Path 770:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
     ...
153. try {
```

[Table of Contents] [Description]

Path 771:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

[Table of Contents] [Description]

Path 772:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
452. try {
```

[Table of Contents] [Description]

Path 773:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
520. try {
```

[Table of Contents] [Description]

Path 774:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
547. try {
```

[Table of Contents] [Description]

Path 775:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      |...
      |559. try {
```

[Table of Contents] [Description]

Path 776:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      |...
      |576. try {
```

[Table of Contents] [Description]

Path 777:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      |...
      |597. try {
```

[Table of Contents] [Description]

Path 778:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      |...
      |617. try {
```

[Table of Contents] [Description]

Path 779:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
```

```
...
68. try {
```

[Table of Contents] [Description]

Path 780:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
...
85. try {
```

[Table of Contents] [Description]

Path 781:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java
...
108. try {
```

[Table of Contents] [Description]

Path 782:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
69. public class CategorySelector extends
FtlCategoryTreeListener //categoryselector.java
...
101. try {
```

[Table of Contents] [Description]

Path 783:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
62. public class CategoryTreeListenerImpl extends
FtlCategoryTreeListener //categorytreelistenerimpl.java
...
87. try {
```

[Table of Contents] [Description]

Path 784:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public class CategoryTreePath extends
FtlCategoryTreeListener //categorytreepath.java
      ...
      93. try {
```

[Table of Contents] [Description]

Path 785:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
70. public void process() //ftlcategorytreelistener.java
      71. try {
```

[Table of Contents] [Description]

Path 786:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public class ListAllForumImpl extends
FtlCategoryTreeListener //listallforumimpl.java
      ...
      86. try {
```

[Table of Contents] [Description]

Path 787:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
66. public class ListForums extends FtlCategoryTreeListener //listforums.java
      ...
      97. try {
```

[Table of Contents] [Description]

Path 788:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public class ModcpImpl extends FtlCategoryTreeListener //modcpimpl.java
      ...
      87. try {
```

[Table of Contents] [Description]

Path 789:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public class RSSImpl extends FtlCategoryTreeListener //rssimpl.java
      ...
87. try {
```

[Table of Contents] [Description]

Path 790:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
62. public class ViewMemberPermissionsImpl extends
FtlCategoryTreeListener //viewmemberpermissionsimpl.java
      ...
77. try {
```

[Table of Contents] [Description]

Path 791:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
131. public static void deleteAttachments_inForum(int forumID) throws
DatabaseException //attachmentutil.java
      ...
147. try {
```

[Table of Contents] [Description]

Path 792:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
138. private void loadSchemaMapping() //membermapping.java
      ...
140. try {
```

[Table of Contents] [Description]

Path 793:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
172. public void writeCaptchaImage(HttpServletRequest request, HttpServletResponse
response) //mvncaptchaservice.java
      ...
181. try {
```

[Table of Contents] [Description]

Path 794:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
172. public void writeCaptchaImage(HttpServletRequest request, HttpServletResponse
response) //mvncaptchaservice.java
      ...
201. try {
```

[Table of Contents] [Description]

Path 795:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void sendActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
      ...
77. try {
```

[Table of Contents] [Description]

Path 796:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
66. public static void updateThreadStatistics(int threadID) //statisticsutil.java
      ...
81. try {
```

[Table of Contents] [Description]

Path 797:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
99. public static void updateForumStatistics(int forumID) //statisticsutil.java
      ...
118. try {
```

[Table of Contents] [Description]

Path 798:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public void run() //watchsendtask.java
      ...
80. try {
```

[Table of Contents] [Description]

Path 799:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
82. public static void addNewMember(HttpServletRequest request, String memberName,
String memberPassword, String memberEmail, Date memberBirthday) //kgutils.java
      ...
155. try {
```

[Table of Contents] [Description]

Path 800:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public CategoryCache() //categorycache.java
      ...
69. try {
```

[Table of Contents] [Description]

Path 801:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
67. public ForumCache() //forumcache.java
      ...
69. try {
```

[Table of Contents] [Description]

Path 802:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
65. public MemberCache() //membercache.java
      ...
67. try {
```

[Table of Contents] [Description]

Path 803:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
69. public PostCache() //postcache.java
      ...
71. try {
```

[Table of Contents] [Description]

Path 804:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
66. public ThreadCache() //threadcache.java
      ...
68. try {
```

[Table of Contents] [Description]

Path 805:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
69. public final String getNameFromID(int
memberID) //memberdaoexternaluserdatabaseabstract.java
      ...
76. try {
```

[Table of Contents] [Description]

Path 806:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void create(int postID, int memberID, String
attachFilename, //attachmentdaoimpljdbc.java
      ...
87. try {
```

[Table of Contents] [Description]

Path 807:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void create(int postID, int memberID, String
attachFilename, //attachmentdaoimpljdbc.java
      ...
97. try {
```

[Table of Contents] [Description]

Path 808:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void create(int postID, int memberID, String
```

```
attachFilename, //attachmentdaoimpljdbc.java
    ...
111. try {
```

[Table of Contents] [Description]

Path 809:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
145. public int createAttachment(int postID, int memberID, String
attachFilename, //attachmentdaoimpljdbc.java
    ...
154. try {
```

[Table of Contents] [Description]

Path 810:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
164. public void delete(int attachID) //attachmentdaoimpljdbc.java
    ...
173. try {
```

[Table of Contents] [Description]

Path 811:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
196. public AttachmentBean getAttachment(int attachID) //attachmentdaoimpljdbc.java
    ...
207. try {
```

[Table of Contents] [Description]

Path 812:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
248. public Collection getAttachments() //attachmentdaoimpljdbc.java
    ...
260. try {
```

[Table of Contents] [Description]

Path 813:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
293. public int getNumberOfAttachments(int category, int
forum) //attachmentdaoimpljdbc.java
     ...
318. try {
```

[Table of Contents] [Description]

Path 814:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public int getNumberOfAttachments_inPost(int
postID) //attachmentdaoimpljdbc.java
     ...
351. try {
```

[Table of Contents] [Description]

Path 815:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
368. public int getNumberOfAttachments_inThread(int
threadID) //attachmentdaoimpljdbc.java
     ...
379. try {
```

[Table of Contents] [Description]

Path 816:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
402. protected static int findAttachID(int postID, int memberID, Timestamp
attachCreationDate) //attachmentdaoimpljdbc.java
     ...
412. try {
```

[Table of Contents] [Description]

Path 817:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
434. public void delete_inPost(int postID) //attachmentdaoimpljdbc.java
     ...
443. try {
```

[Table of Contents] [Description]

Path 818:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
464. public Collection getAttachments_inPost(int
postID) //attachmentdaoimpljdbc.java
      |...
      |476. try {
```

[Table of Contents] [Description]

Path 819:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
515. public Collection getAttachments_inThread(int
threadID) //attachmentdaoimpljdbc.java
      |...
      |527. try {
```

[Table of Contents] [Description]

Path 820:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
566. public Collection getAttachments_inForum(int
forumID) //attachmentdaoimpljdbc.java
      |...
      |578. try {
```

[Table of Contents] [Description]

Path 821:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
614. public void increaseDownloadCount(int attachID) //attachmentdaoimpljdbc.java
      |...
      |620. try {
```

[Table of Contents] [Description]

Path 822:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
639. public void updateAttachDesc(int attachID, String
attachDesc) //attachmentdaoimpljdbc.java
      |
```

```
...
645. try {
```

[Table of Contents] [Description]

Path 823:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
670. public void updateAttachOption(int attachID, int
attachOption) //attachmentdaoimpljdbc.java
...
676. try {
```

[Table of Contents] [Description]

Path 824:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
720. public Collection getAttachments_withSortSupport_limit_mysql(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
...
745. try {
```

[Table of Contents] [Description]

Path 825:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
788. public Collection getAttachments_withSortSupport_limit_general(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
...
812. try {
```

[Table of Contents] [Description]

Path 826:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
788. public Collection getAttachments_withSortSupport_limit_general(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
...
824. try {
```

[Table of Contents] [Description]

Path 827:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
863. public int getMaxAttachmentID() throws
DatabaseException //attachmentdaoimpljdbc.java
      |...
      |871. try {
```

[Table of Contents] [Description]

Path 828:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
887. public Collection getAttachments_fromIDRange(int fromID, int
toID) //attachmentdaoimpljdbc.java
      |...
      |904. try {
```

[Table of Contents] [Description]

Path 829:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public int findByPrimaryKey(int categoryID) //categorydaoimpljdbc.java
      |...
      |85. try {
```

[Table of Contents] [Description]

Path 830:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public int findByAlternateKey_CategoryName(String
categoryName) //categorydaoimpljdbc.java
      |...
      |119. try {
```

[Table of Contents] [Description]

Path 831:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. public void create(int parentCategoryID, String categoryName, String
categoryDesc, //categorydaoimpljdbc.java
      |...
      |151. try {
```

[Table of Contents] [Description]

Path 832:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. public void create(int parentCategoryID, String categoryName, String
categoryDesc, //categorydaoimpljdbc.java
     ...
160. try {
```

[Table of Contents] [Description]

Path 833:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. public void create(int parentCategoryID, String categoryName, String
categoryDesc, //categorydaoimpljdbc.java
     ...
175. try {
```

[Table of Contents] [Description]

Path 834:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
205. public void delete(int categoryID) //categorydaoimpljdbc.java
     ...
214. try {
```

[Table of Contents] [Description]

Path 835:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
236. public void update(int categoryID, // primary key //categorydaoimpljdbc.java
     ...
245. try {
```

[Table of Contents] [Description]

Path 836:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
236. public void update(int categoryID, // primary key //categorydaoimpljdbc.java
     ...
```

```
258. try {
```

[Table of Contents] [Description]

Path 837:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
295. public CategoryBean getCategory(int categoryID) //categorydaoimpljdbc.java
     ...
305. try {
```

[Table of Contents] [Description]

Path 838:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public Collection getCategories() //categorydaoimpljdbc.java
     ...
353. try {
```

[Table of Contents] [Description]

Path 839:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
388. public void decreaseCategoryOrder(int categoryID, Timestamp
categoryModifiedDate) //categorydaoimpljdbc.java
     ...
394. try {
```

[Table of Contents] [Description]

Path 840:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
417. public void increaseCategoryOrder(int categoryID, Timestamp
categoryModifiedDate) //categorydaoimpljdbc.java
     ...
423. try {
```

[Table of Contents] [Description]

Path 841:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public void findByPrimaryKey(int memberID, int
threadID) //favoritethreaddaoimpljdbc.java
      ...
83. try {
```

[Table of Contents] [Description]

Path 842:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
      ...
115. try {
```

[Table of Contents] [Description]

Path 843:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
      ...
124. try {
```

[Table of Contents] [Description]

Path 844:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
      ...
132. try {
```

[Table of Contents] [Description]

Path 845:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
      ...
140. try {
```

[Table of Contents] [Description]

Path 846:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
      ...
153. try {
```

[Table of Contents] [Description]

Path 847:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
178. public void delete(int memberID, int threadID) //favoritethreaddaoimpljdbc.java
      ...
187. try {
```

[Table of Contents] [Description]

Path 848:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
205. public void delete_inThread(int threadID) //favoritethreaddaoimpljdbc.java
      ...
214. try {
```

[Table of Contents] [Description]

Path 849:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
230. public void delete_inForum(int forumID) //favoritethreaddaoimpljdbc.java
      ...
239. try {
```

[Table of Contents] [Description]

Path 850:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
255. public void delete_inMember(int memberID) //favoritethreaddaoimpljdbc.java
      ...
264. try {
```

[Table of Contents] [Description]

Path 851:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
282. public void update_ForumID_inThread(int threadID, int
forumID) //favoritethreaddaoimpljdbc.java
     ...
285. try {
```

[Table of Contents] [Description]

Path 852:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
282. public void update_ForumID_inThread(int threadID, int
forumID) //favoritethreaddaoimpljdbc.java
     ...
298. try {
```

[Table of Contents] [Description]

Path 853:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
324. public Collection getFavoriteThreads_inMember(int
memberID) //favoritethreaddaoimpljdbc.java
     ...
336. try {
```

[Table of Contents] [Description]

Path 854:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
390. public int getNumberOfFavoriteThreads_inMember(int
memberID) //favoritethreaddaoimpljdbc.java
     ...
400. try {
```

[Table of Contents] [Description]

Path 855:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. public void findByPrimaryKey(int forumID) //forumdaoimpljdbc.java
      ...
84. try {
```

[Table of Contents] [Description]

Path 856:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
102. public void findByAlternateKey_ForumName_CategoryID(String forumName, int
categoryID) //forumdaoimpljdbc.java
      ...
116. try {
```

[Table of Contents] [Description]

Path 857:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
      ...
159. try {
```

[Table of Contents] [Description]

Path 858:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
      ...
168. try {
```

[Table of Contents] [Description]

Path 859:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
      ...
176. try {
```

[Table of Contents] [Description]

Path 860:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
     ...
191. try {
```

[Table of Contents] [Description]

Path 861:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public int createForum(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
     ...
249. try {
```

[Table of Contents] [Description]

Path 862:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
257. public void delete(int forumID) //forumdaoimpljdbc.java
     ...
266. try {
```

[Table of Contents] [Description]

Path 863:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. public void update(int forumID, // primary key //forumdaoimpljdbc.java
     ...
308. try {
```

[Table of Contents] [Description]

Path 864:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. public void update(int forumID, // primary key //forumdaoimpljdbc.java
     ...
316. try {
```

[Table of Contents] [Description]

Path 865:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. public void update(int forumID, // primary key //forumdaoimpljdbc.java
       ...
324. try {
```

[Table of Contents] [Description]

Path 866:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. public void update(int forumID, // primary key //forumdaoimpljdbc.java
       ...
339. try {
```

[Table of Contents] [Description]

Path 867:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
383. public void updateLastPostMemberName(int forumID, // primary
key //forumdaoimpljdbc.java
       ...
389. try {
```

[Table of Contents] [Description]

Path 868:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
383. public void updateLastPostMemberName(int forumID, // primary
key //forumdaoimpljdbc.java
       ...
403. try {
```

[Table of Contents] [Description]

Path 869:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
442. public void updateLastPostDate(int forumID, // primary
key //forumdaoimpljdbc.java
```

```
...
451. try {
```

[Table of Contents] [Description]

Path 870:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
474. public void resetForumOwnerNameWhenDeleteMember(String
memberName) //forumdaoimpljdbc.java
...
490. try {
```

[Table of Contents] [Description]

Path 871:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
510. public void updateStatistics(int forumID, // primary
key //forumdaoimpljdbc.java
...
519. try {
```

[Table of Contents] [Description]

Path 872:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
546. public void increasePostCount(int forumID) //forumdaoimpljdbc.java
...
552. try {
```

[Table of Contents] [Description]

Path 873:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
574. public void increaseThreadCount(int forumID) //forumdaoimpljdbc.java
...
580. try {
```

[Table of Contents] [Description]

Path 874:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
600. public void decreaseThreadCount(int forumID) //forumdaoimpljdbc.java
       ...
606. try {
```

[Table of Contents] [Description]

Path 875:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
630. public ForumBean getForum(int forumID) //forumdaoimpljdbc.java
       ...
640. try {
```

[Table of Contents] [Description]

Path 876:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
687. public Collection getForums() //forumdaoimpljdbc.java
       ...
699. try {
```

[Table of Contents] [Description]

Path 877:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
743. public Collection getForums_inCategory(int categoryID) //forumdaoimpljdbc.java
       ...
755. try {
```

[Table of Contents] [Description]

Path 878:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
800. public void decreaseForumOrder(int forumID, Timestamp
forumModifiedDate) //forumdaoimpljdbc.java
       ...
806. try {
```

[Table of Contents] [Description]

Path 879:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
829. public void increaseForumOrder(int forumID, Timestamp
forumModifiedDate) //forumdaoimpljdbc.java
     ...
     835. try {
```

[Table of Contents] [Description]

Path 880:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
862. protected static ForumBean getBean_byAlternateKey_ForumName_CategoryID(String
forumName, int categoryID) //forumdaoimpljdbc.java
     ...
     872. try {
```

[Table of Contents] [Description]

Path 881:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
     ...
     945. try {
```

[Table of Contents] [Description]

Path 882:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
     ...
     950. try {
```

[Table of Contents] [Description]

Path 883:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
     ...
```

```
1024. try {
```

[Table of Contents] [Description]

Path 884:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
...
1031. try {
```

[Table of Contents] [Description]

Path 885:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
...
1106. try {
```

[Table of Contents] [Description]

Path 886:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
...
1111. try {
```

[Table of Contents] [Description]

Path 887:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
...
1212. try {
```

[Table of Contents] [Description]

Path 888:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
1217. try {
```

[Table of Contents] [Description]

Path 889:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
1294. try {
```

[Table of Contents] [Description]

Path 890:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
1299. try {
```

[Table of Contents] [Description]

Path 891:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
71. public void findByPrimaryKey(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
      ...
81. try {
```

[Table of Contents] [Description]

Path 892:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int groupID, int forumID, int
```

```
permission) //groupforumdaoimpljdbc.java
      ...
      111. try {
```

[Table of Contents] [Description]

Path 893:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
      ...
      120. try {
```

[Table of Contents] [Description]

Path 894:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
      ...
      128. try {
```

[Table of Contents] [Description]

Path 895:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
      ...
      141. try {
```

[Table of Contents] [Description]

Path 896:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
162. public void delete(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
      ...
      171. try {
```

[Table of Contents] [Description]

Path 897:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
190. public void delete_inGroup(int groupID) //groupforumdaoimpljdbc.java
       ...
199. try {
```

[Table of Contents] [Description]

Path 898:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
215. public void delete_inForum(int forumID) //groupforumdaoimpljdbc.java
       ...
224. try {
```

[Table of Contents] [Description]

Path 899:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
249. public Collection getBeans_inGroupForum(int groupID, int
forumID) //groupforumdaoimpljdbc.java
       ...
261. try {
```

[Table of Contents] [Description]

Path 900:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
290. public Collection getBeans_inForum(int forumID) //groupforumdaoimpljdbc.java
       ...
302. try {
```

[Table of Contents] [Description]

Path 901:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
330. public Collection getBeans_inGroup(int groupID) //groupforumdaoimpljdbc.java
       ...
343. try {
```

[Table of Contents] [Description]

Path 902:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
371. public Collection getDistinctGroups() //groupforumdaoimpljdbc.java
       ...
382. try {
```

[Table of Contents] [Description]

Path 903:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
71. public void findByPrimaryKey(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
       ...
81. try {
```

[Table of Contents] [Description]

Path 904:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
       ...
110. try {
```

[Table of Contents] [Description]

Path 905:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
       ...
119. try {
```

[Table of Contents] [Description]

Path 906:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
```

```
...
132. try {
```

[Table of Contents] [Description]

Path 907:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
152. public void delete(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
...
161. try {
```

[Table of Contents] [Description]

Path 908:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public void delete_inGroup(int groupID) //grouppermissiondaoimpljdbc.java
...
188. try {
```

[Table of Contents] [Description]

Path 909:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
214. public Collection getBeans_inGroup(int
groupID) //grouppermissiondaoimpljdbc.java
...
226. try {
```

[Table of Contents] [Description]

Path 910:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
253. public Collection getDistinctGroups() //grouppermissiondaoimpljdbc.java
...
264. try {
```

[Table of Contents] [Description]

Path 911:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. public void findByPrimaryKey(int groupID) //groupsdaoimpljdbc.java
      ...
84. try {
```

[Table of Contents] [Description]

Path 912:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
102. public void findByAlternateKey_GroupName(String
groupName) //groupsdaoimpljdbc.java
      ...
116. try {
```

[Table of Contents] [Description]

Path 913:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
      ...
152. try {
```

[Table of Contents] [Description]

Path 914:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
      ...
161. try {
```

[Table of Contents] [Description]

Path 915:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
      ...
167. try {
```

[Table of Contents] [Description]

Path 916:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
      ...
183. try {
```

[Table of Contents] [Description]

Path 917:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
212. public void delete(int groupID) //groupsdaoimpljdbc.java
      ...
221. try {
```

[Table of Contents] [Description]

Path 918:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
242. public void update(int groupID, // primary key //groupsdaoimpljdbc.java
      ...
510. public int getGroupIDFromGroupName(String groupName)
```

[Table of Contents] [Description]

Path 919:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
242. public void update(int groupID, // primary key //groupsdaoimpljdbc.java
      ...
263. try {
```

[Table of Contents] [Description]

Path 920:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. public void updateOwner(int groupID, // primary key //groupsdaoimpljdbc.java
      ...
304. try {
```

[Table of Contents] [Description]

Path 921:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. public void updateOwner(int groupID, // primary key //groupsdaoimpljdbc.java
       ...
310. try {
```

[Table of Contents] [Description]

Path 922:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
296. public void updateOwner(int groupID, // primary key //groupsdaoimpljdbc.java
       ...
326. try {
```

[Table of Contents] [Description]

Path 923:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
356. public GroupsBean getGroup(int groupID) //groupsdaoimpljdbc.java
       ...
366. try {
```

[Table of Contents] [Description]

Path 924:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
395. public Collection getMyGroups(int memberID) //groupsdaoimpljdbc.java
       ...
408. try {
```

[Table of Contents] [Description]

Path 925:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
439. public Collection getGroups() //groupsdaoimpljdbc.java
       ...
452. try {
```

[Table of Contents] [Description]

Path 926:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
479. public int getNumberOfGroups() //groupsdaoimpljdbc.java
     ...
489. try {
```

[Table of Contents] [Description]

Path 927:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
510. public int getGroupIDFromGroupName(String groupName) //groupsdaoimpljdbc.java
     ...
517. try {
```

[Table of Contents] [Description]

Path 928:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
78. public void findByPrimaryKey(int memberID) //memberdaoimpljdbc.java
    ...
88. try {
```

[Table of Contents] [Description]

Path 929:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. public void findByPrimaryKey2(int memberID, String
memberName) //memberdaoimpljdbc.java
     ...
119. try {
```

[Table of Contents] [Description]

Path 930:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
     ...
155. try {
```

[Table of Contents] [Description]

Path 931:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
177. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
     ...
191. try {
```

[Table of Contents] [Description]

Path 932:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
219. public void deleteByPrimaryKey(int memberID) throws
DatabaseException //memberdaoimpljdbc.java
     ...
228. try {
```

[Table of Contents] [Description]

Path 933:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
     ...
280. try {
```

[Table of Contents] [Description]

Path 934:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
     ...
291. try {
```

[Table of Contents] [Description]

Path 935:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
      ...
309. try {
```

[Table of Contents] [Description]

Path 936:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
394. public void update(int memberID, // primary key //memberdaoimpljdbc.java
      ...
412. try {
```

[Table of Contents] [Description]

Path 937:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
      ...
488. try {
```

[Table of Contents] [Description]

Path 938:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
      ...
501. try {
```

[Table of Contents] [Description]

Path 939:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
554. try {
```

[Table of Contents] [Description]

Path 940:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
602. public void updateTempPassword(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
611. try {
```

[Table of Contents] [Description]

Path 941:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
659. try {
```

[Table of Contents] [Description]

Path 942:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
698. public void updateAvatar(int memberID, // primary key //memberdaoimpljdbc.java
     ...
707. try {
```

[Table of Contents] [Description]

Path 943:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
746. public void updateSignature(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
755. try {
```

[Table of Contents] [Description]

Path 944:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
794. public void updateTitle(int memberID, // primary key //memberdaoimpljdbc.java
     ...
803. try {
```

[Table of Contents] [Description]

Path 945:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
842. public void updateLastLogon(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
851. try {
```

[Table of Contents] [Description]

Path 946:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
      ...
901. try {
```

[Table of Contents] [Description]

Path 947:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
949. try {
```

[Table of Contents] [Description]

Path 948:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
      ...
996. try {
```

[Table of Contents] [Description]

Path 949:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
      ...
1217. try {
```

[Table of Contents] [Description]

Path 950:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1297. public int getNumberOfMembers() //memberdaoimpljdbc.java
      |...
      1307. try {
```

[Table of Contents] [Description]

Path 951:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1331. public int getNumberOfMembers_inMemberStatus(int
memberStatus) //memberdaoimpljdbc.java
      |...
      1341. try {
```

[Table of Contents] [Description]

Path 952:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1366. public int getNumberOfMembers_inActivationStatus(boolean
activated) //memberdaoimpljdbc.java
      |...
      1381. try {
```

[Table of Contents] [Description]

Path 953:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      |...
      1413. try {
```

[Table of Contents] [Description]

Path 954:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
      |
```

```
...
1443. try {
```

[Table of Contents] [Description]

Path 955:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1544. try {
```

[Table of Contents] [Description]

Path 956:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1674. try {
```

[Table of Contents] [Description]

Path 957:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1809. try {
```

[Table of Contents] [Description]

Path 958:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1813. try {
```

[Table of Contents] [Description]

Path 959:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1900. public Collection getEnableMembers_inActivationStatus(String
kind) //memberdaoimpljdbc.java
      ...
1922. try {
```

[Table of Contents] [Description]

Path 960:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
      ...
2006. try {
```

[Table of Contents] [Description]

Path 961:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2032. public void updatePostCount(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
2041. try {
```

[Table of Contents] [Description]

Path 962:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2070. public void increaseViewCount(int memberID) //memberdaoimpljdbc.java
      ...
2076. try {
```

[Table of Contents] [Description]

Path 963:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2101. public void increasePostCount(int memberID) //memberdaoimpljdbc.java
      ...
2107. try {
```

[Table of Contents] [Description]

Path 964:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
      ...
2169. try {
```

[Table of Contents] [Description]

Path 965:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
      ...
2174. try {
```

[Table of Contents] [Description]

Path 966:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2207. public int getNumberOfMembers_inExpire(Timestamp
expireDate) //memberdaoimpljdbc.java
      ...
2217. try {
```

[Table of Contents] [Description]

Path 967:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2237. public void updateMember_expireDate(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
2246. try {
```

[Table of Contents] [Description]

Path 968:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2272. public Collection getMembers() //memberdaoimpljdbc.java
      ...
```

```
2283. try {
```

[Table of Contents] [Description]

Path 969:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2343. public int getMaxMemberID() throws DatabaseException //memberdaoimpljdbc.java
      ...
2351. try {
```

[Table of Contents] [Description]

Path 970:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2370. public Collection getMembers_fromIDRange(int fromID, int toID) throws
IllegalArgumentException, DatabaseException //memberdaoimpljdbc.java
      ...
2384. try {
```

[Table of Contents] [Description]

Path 971:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2446. public Collection getEnableMembers_inGroup(int
groupID) //memberdaoimpljdbc.java
      ...
2458. try {
```

[Table of Contents] [Description]

Path 972:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2520. public Collection getForumsAuthorizedMembers() throws
DatabaseException //memberdaoimpljdbc.java
      ...
2532. try {
```

[Table of Contents] [Description]

Path 973:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2561. public Collection getAuthorizedMembers() throws
DatabaseException //memberdaoimpljdbc.java
      ...
2573. try {
```

[Table of Contents] [Description]

Path 974:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2602. public Collection getNonActivatedNoPostMembers(Timestamp
before) //memberdaoimpljdbc.java
      ...
2617. try {
```

[Table of Contents] [Description]

Path 975:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
71. public void findByPrimaryKey(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
81. try {
```

[Table of Contents] [Description]

Path 976:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
111. try {
```

[Table of Contents] [Description]

Path 977:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
121. try {
```

[Table of Contents] [Description]

Path 978:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
       ...
130. try {
```

[Table of Contents] [Description]

Path 979:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
       ...
143. try {
```

[Table of Contents] [Description]

Path 980:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
164. public void delete(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
       ...
173. try {
```

[Table of Contents] [Description]

Path 981:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
192. public void delete_inMember(int memberID) //memberforumdaoimpljdbc.java
       ...
201. try {
```

[Table of Contents] [Description]

Path 982:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
216. public void delete_inForum(int forumID) //memberforumdaoimpljdbc.java
       ...
```

```
225. try {
```

[Table of Contents] [Description]

Path 983:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
245. public Collection getBeans_inMemberForum(int memberID, int
forumID) //memberforumdaoimpljdbc.java
     ...
256. try {
```

[Table of Contents] [Description]

Path 984:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
285. public Collection getBeans_inMember(int memberID) //memberforumdaoimpljdbc.java
     ...
297. try {
```

[Table of Contents] [Description]

Path 985:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
324. public Collection getBeans_inForum(int forumID) //memberforumdaoimpljdbc.java
     ...
338. try {
```

[Table of Contents] [Description]

Path 986:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public void findByPrimaryKey(int groupID, int
memberID) //membergroupdaoimpljdbc.java
    ...
83. try {
```

[Table of Contents] [Description]

Path 987:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int groupID, String
memberName, //membergroupdaoimpljdbc.java
      ...
112. try {
```

[Table of Contents] [Description]

Path 988:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int groupID, String
memberName, //membergroupdaoimpljdbc.java
      ...
124. try {
```

[Table of Contents] [Description]

Path 989:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int groupID, String
memberName, //membergroupdaoimpljdbc.java
      ...
133. try {
```

[Table of Contents] [Description]

Path 990:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(int groupID, String
memberName, //membergroupdaoimpljdbc.java
      ...
146. try {
```

[Table of Contents] [Description]

Path 991:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void delete(int groupID, int memberID) //membergroupdaoimpljdbc.java
      ...
179. try {
```

[Table of Contents] [Description]

Path 992:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
197. public void delete_inGroup(int groupID) //membergroupdaoimpljdbc.java
     ...
206. try {
```

[Table of Contents] [Description]

Path 993:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. public void delete_inMember(int memberID) //membergroupdaoimpljdbc.java
     ...
231. try {
```

[Table of Contents] [Description]

Path 994:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
254. public Collection getBeans_inGroup(int groupID) //membergroupdaoimpljdbc.java
     ...
266. try {
```

[Table of Contents] [Description]

Path 995:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
292. public int getNumberOfBeans_inGroup(int groupID) //membergroupdaoimpljdbc.java
     ...
302. try {
```

[Table of Contents] [Description]

Path 996:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
326. public Collection getBeans_limit(int groupID, int offset, int
rowsToReturn) //membergroupdaoimpljdbc.java
     ...
341. try {
```

[Table of Contents] [Description]

Path 997:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
71. public void findByPrimaryKey(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
81. try {
```

[Table of Contents] [Description]

Path 998:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
110. try {
```

[Table of Contents] [Description]

Path 999:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
120. try {
```

[Table of Contents] [Description]

Path 1000:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
134. try {
```

[Table of Contents] [Description]

Path 1001:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. public void delete(int memberID, int
```

```
permission) //memberpermissiondaoimpljdbc.java
    ...
163. try {
```

[Table of Contents] [Description]

Path 1002:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public void delete_inMember(int memberID) //memberpermissiondaoimpljdbc.java
    ...
190. try {
```

[Table of Contents] [Description]

Path 1003:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public Collection getBeans_inMember(int
memberID) //memberpermissiondaoimpljdbc.java
    ...
220. try {
```

[Table of Contents] [Description]

Path 1004:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. private int findMessageID(int memberID, int messageSenderID, Timestamp
messageCreationDate) //messagedaoimpljdbc.java
    ...
85. try {
```

[Table of Contents] [Description]

Path 1005:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void findByPrimaryKey(int messageID) //messagedaoimpljdbc.java
    ...
117. try {
```

[Table of Contents] [Description]

Path 1006:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
     ...
152. try {
```

[Table of Contents] [Description]

Path 1007:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
     ...
160. try {
```

[Table of Contents] [Description]

Path 1008:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
     ...
168. try {
```

[Table of Contents] [Description]

Path 1009:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
     ...
181. try {
```

[Table of Contents] [Description]

Path 1010:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
     ...
213. try {
```

[Table of Contents] [Description]

Path 1011:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public int create(String folderName, int memberID, int
messageSenderID, //messagedaoimpljdbc.java
      |...
      |218. try {
```

[Table of Contents] [Description]

Path 1012:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
295. public Collection getBeans_inMember_inFolder_withSortSupport_limit_mysql(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      |...
      |315. try {
```

[Table of Contents] [Description]

Path 1013:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
364. public Collection getBeans_inMember_inFolder_withSortSupport_limit_noscroll(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      |...
      |383. try {
```

[Table of Contents] [Description]

Path 1014:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
436. public Collection getBeans_inMember_inFolder_withSortSupport_limit_general(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      |...
      |455. try {
```

[Table of Contents] [Description]

Path 1015:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
436. public Collection getBeans_inMember_inFolder_withSortSupport_limit_general(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
     ...
461. try {
```

[Table of Contents] [Description]

Path 1016:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
512. public Collection getPublicMessages() //messagedaoimpljdbc.java
     ...
527. try {
```

[Table of Contents] [Description]

Path 1017:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java
     ...
583. try {
```

[Table of Contents] [Description]

Path 1018:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
625. public int getNumberOfNonPublicMessages_inMember(int
memberID) //messagedaoimpljdbc.java
     ...
639. try {
```

[Table of Contents] [Description]

Path 1019:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
656. public void updateMessageReadStatus(int messageID, // primary
key //messagedaoimpljdbc.java
     ...
666. try {
```

[Table of Contents] [Description]

Path 1020:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
691. public void updateAttachCount(int messageID, int
messageAttachCount) //messagedaoimpljdbc.java
     ...
699. try {
```

[Table of Contents] [Description]

Path 1021:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
722. public void updateFolderName(int messageID, // primary
key //messagedaoimpljdbc.java
     ...
731. try {
```

[Table of Contents] [Description]

Path 1022:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
754. public void deleteMessage(int messageID, int
memberID) //messagedaoimpljdbc.java
     ...
764. try {
```

[Table of Contents] [Description]

Path 1023:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
782. public void deleteSenderMessages(int senderID) //messagedaoimpljdbc.java
     ...
791. try {
```

[Table of Contents] [Description]

Path 1024:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
807. public void deleteMessages_inFolderName_inMember(String folderName, int
memberID) //messagedaoimpljdbc.java
     |
```

```
...
817. try {
```

[Table of Contents] [Description]

Path 1025:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
862. private int getNumberOfMessages_inMember_inFolder(int memberID, String
folderName, boolean unread, boolean onlyNonPublic) //messagedaoimpljdbc.java
...
878. try {
```

[Table of Contents] [Description]

Path 1026:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public void findByPrimaryKey(String folderName, int
memberID) //messagefolderdaoimpljdbc.java
...
83. try {
```

[Table of Contents] [Description]

Path 1027:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(String folderName, int memberID, int
folderOrder, //messagefolderdaoimpljdbc.java
...
115. try {
```

[Table of Contents] [Description]

Path 1028:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(String folderName, int memberID, int
folderOrder, //messagefolderdaoimpljdbc.java
...
124. try {
```

[Table of Contents] [Description]

Path 1029:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void create(String folderName, int memberID, int
folderOrder, //messagefolderdaoimpljdbc.java
     ...
137. try {
```

[Table of Contents] [Description]

Path 1030:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
163. public void delete(String folderName, int
memberID) //messagefolderdaoimpljdbc.java
     ...
172. try {
```

[Table of Contents] [Description]

Path 1031:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
195. public MessageFolderBean getMessageFolder(String folderName, int
memberID) //messagefolderdaoimpljdbc.java
     ...
205. try {
```

[Table of Contents] [Description]

Path 1032:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public Collection getMessageFolders_inMember(int
memberID) //messagefolderdaoimpljdbc.java
     ...
256. try {
```

[Table of Contents] [Description]

Path 1033:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public int getMaxFolderOrder(int memberID) //messagefolderdaoimpljdbc.java
     ...
296. try {
```

[Table of Contents] [Description]

Path 1034:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
318. public void increaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
      |...
      |327. try {
```

[Table of Contents] [Description]

Path 1035:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
356. public void decreaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
      |...
      |365. try {
```

[Table of Contents] [Description]

Path 1036:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public void create(int fromID, int toID, Timestamp
messageCreationDate, //messagestatisticsdaoimpljdbc.java
      |...
      |81. try {
```

[Table of Contents] [Description]

Path 1037:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public void create(int fromID, int toID, Timestamp
messageCreationDate, //messagestatisticsdaoimpljdbc.java
      |...
      |89. try {
```

[Table of Contents] [Description]

Path 1038:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public void create(int fromID, int toID, Timestamp
messageCreationDate, //messagestatisticsdaoimpljdbc.java
      ...
102. try {
```

[Table of Contents] [Description]

Path 1039:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
127. public void delete_inMember(int memberID) //messagestatisticsdaoimpljdbc.java
      ...
136. try {
```

[Table of Contents] [Description]

Path 1040:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
152. public int getNumberOfBeans_inFromID(int
fromID) //messagestatisticsdaoimpljdbc.java
      ...
162. try {
```

[Table of Contents] [Description]

Path 1041:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
179. public int getNumberOfBeans_inToID(int
toID) //messagestatisticsdaoimpljdbc.java
      ...
189. try {
```

[Table of Contents] [Description]

Path 1042:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
206. public int getNumberOfBeans_inFromID_supportTimestamp(int fromID, Timestamp
from, Timestamp to) //messagestatisticsdaoimpljdbc.java
      ...
219. try {
```

[Table of Contents] [Description]

Path 1043:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
238. public int getNumberOfBeans_inToID_supportTimestamp(int toID, Timestamp from,
Timestamp to) //messagestatisticsdaoimpljdbc.java
     ...
251. try {
```

[Table of Contents] [Description]

Path 1044:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. protected static int findPmAttachID(int memberID, Timestamp
pmAttachCreationDate) //pmattachmentdaoimpljdbc.java
     ...
86. try {
```

[Table of Contents] [Description]

Path 1045:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
107. public void findByPrimaryKey(int pmAttachID) //pmattachmentdaoimpljdbc.java
     ...
117. try {
```

[Table of Contents] [Description]

Path 1046:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
141. public int create(int memberID, String pmAttachFilename, int
pmAttachFileSize, //pmattachmentdaoimpljdbc.java
     ...
147. try {
```

[Table of Contents] [Description]

Path 1047:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
141. public int create(int memberID, String pmAttachFilename, int
pmAttachFileSize, //pmattachmentdaoimpljdbc.java
     ...
```

```
160. try {
```

[Table of Contents] [Description]

Path 1048:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
141. public int create(int memberID, String pmAttachFilename, int
pmAttachFileSize, //pmattachmentdaoimpljdbc.java
...
193. try {
```

[Table of Contents] [Description]

Path 1049:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
203. public void delete(int pmAttachID) //pmattachmentdaoimpljdbc.java
...
212. try {
```

[Table of Contents] [Description]

Path 1050:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
...
245. try {
```

[Table of Contents] [Description]

Path 1051:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
283. public void increaseDownloadCount(int
pmAttachID) //pmattachmentdaoimpljdbc.java
...
289. try {
```

[Table of Contents] [Description]

Path 1052:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
314. public Collection getPmAttachments_inMessage(int
messageID) //pmattachmentdaoimpljdbc.java
      ...
326. try {
```

[Table of Contents] [Description]

Path 1053:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
365. public Collection getOrphanPmAttachments() //pmattachmentdaoimpljdbc.java
      ...
386. try {
```

[Table of Contents] [Description]

Path 1054:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
417. public void updatePmAttachOption(int pmAttachID, int
pmAttachOption) //pmattachmentdaoimpljdbc.java
      ...
423. try {
```

[Table of Contents] [Description]

Path 1055:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public void findByPrimaryKey(int messageID, int
pmAttachID) //pmattachmessagedaoimpljdbc.java
      ...
83. try {
```

[Table of Contents] [Description]

Path 1056:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public void create(int messageID, int pmAttachID, int
relationType, //pmattachmessagedaoimpljdbc.java
      ...
113. try {
```

[Table of Contents] [Description]

Path 1057:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public void create(int messageID, int pmAttachID, int
relationType, //pmattachmessagedaoimpljdbc.java
     ...
122. try {
```

[Table of Contents] [Description]

Path 1058:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public void create(int messageID, int pmAttachID, int
relationType, //pmattachmessagedaoimpljdbc.java
     ...
130. try {
```

[Table of Contents] [Description]

Path 1059:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public void create(int messageID, int pmAttachID, int
relationType, //pmattachmessagedaoimpljdbc.java
     ...
143. try {
```

[Table of Contents] [Description]

Path 1060:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
166. public void delete(int messageID, int
pmAttachID) //pmattachmessagedaoimpljdbc.java
     ...
175. try {
```

[Table of Contents] [Description]

Path 1061:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
193. public void delete_inMessage(int messageID) //pmattachmessagedaoimpljdbc.java
     ...
```

```
202. try {
```

[Table of Contents] [Description]

Path 1062:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
221. public Collection getBeans_inMessage(int
messageID) //pmattachmessagedaoimpljdbc.java
...
233. try {
```

[Table of Contents] [Description]

Path 1063:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
262. public Collection getBeans_inPmAttach(int
pmAttachID) //pmattachmessagedaoimpljdbc.java
...
274. try {
```

[Table of Contents] [Description]

Path 1064:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
299. public int getNumberOfBeans_inMessage(int
messageID) //pmattachmessagedaoimpljdbc.java
...
309. try {
```

[Table of Contents] [Description]

Path 1065:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
326. public int getNumberOfBeans_inPmAttach(int
pmAttachID) //pmattachmessagedaoimpljdbc.java
...
336. try {
```

[Table of Contents] [Description]

Path 1066:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public void findByPrimaryKey(int postID) //postdaoimpljdbc.java
      ...
85. try {
```

[Table of Contents] [Description]

Path 1067:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
118. try {
```

[Table of Contents] [Description]

Path 1068:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
128. try {
```

[Table of Contents] [Description]

Path 1069:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
137. try {
```

[Table of Contents] [Description]

Path 1070:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
147. try {
```

[Table of Contents] [Description]

Path 1071:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
     ...
160. try {
```

[Table of Contents] [Description]

Path 1072:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
     ...
169. try {
```

[Table of Contents] [Description]

Path 1073:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
     ...
184. try {
```

[Table of Contents] [Description]

Path 1074:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
224. public int createPost(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
     ...
239. try {
```

[Table of Contents] [Description]

Path 1075:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
224. public int createPost(int parentPostID, int forumID, int
```

```
threadID, //postdaoimpljdbc.java
    ...
244. try {
```

[Table of Contents] [Description]

Path 1076:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
253. public void delete(int postID) //postdaoimpljdbc.java
    ...
262. try {
```

[Table of Contents] [Description]

Path 1077:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
279. public void delete_inThread(int threadID) //postdaoimpljdbc.java
    ...
288. try {
```

[Table of Contents] [Description]

Path 1078:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
305. public void delete_inForum(int forumID) //postdaoimpljdbc.java
    ...
314. try {
```

[Table of Contents] [Description]

Path 1079:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
335. public void update(int postID, // primary key //postdaoimpljdbc.java
    ...
343. try {
```

[Table of Contents] [Description]

Path 1080:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
335. public void update(int postID, // primary key //postdaoimpljdbc.java
       ...
359. try {
```

[Table of Contents] [Description]

Path 1081:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
400. public void update(int postID, // primary key //postdaoimpljdbc.java
       ...
409. try {
```

[Table of Contents] [Description]

Path 1082:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
441. public void updateAttachCount(int postID, // primary key //postdaoimpljdbc.java
       ...
450. try {
```

[Table of Contents] [Description]

Path 1083:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
480. public void updateStatus(int postID, // primary key //postdaoimpljdbc.java
       ...
489. try {
```

[Table of Contents] [Description]

Path 1084:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
515. public void update_ForumID_inThread(int threadID, int
forumID) //postdaoimpljdbc.java
       ...
518. try {
```

[Table of Contents] [Description]

Path 1085:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- Information

```
515. public void update_ForumID_inThread(int threadID, int
forumID) //postdaoimpljdbc.java
     ...
     531. try {
```

[Table of Contents] [Description]

Path 1086:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
552. private int findPostID(int forumID, String memberName, Timestamp
postCreationDate) //postdaoimpljdbc.java
     ...
     562. try {
```

[Table of Contents] [Description]

Path 1087:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
591. public PostBean getPost(int postID) //postdaoimpljdbc.java
     ...
     601. try {
```

[Table of Contents] [Description]

Path 1088:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
683. private Collection getBeans_inThread_limit_mysql(int threadID, int offset, int
rowsToReturn, boolean enable) //postdaoimpljdbc.java
     ...
     703. try {
```

[Table of Contents] [Description]

Path 1089:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
751. private Collection getBeans_inThread_limit_noscroll(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
     ...
     770. try {
```

[Table of Contents] [Description]

Path 1090:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
     |...
     |841. try {
```

[Table of Contents] [Description]

Path 1091:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
     |...
     |845. try {
```

[Table of Contents] [Description]

Path 1092:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
891. public int getNumberOfPosts_inMember(int memberID) //postdaoimpljdbc.java
     |...
     |901. try {
```

[Table of Contents] [Description]

Path 1093:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
932. public int getNumberOfPosts_inForum(int forumID, boolean
enable) //postdaoimpljdbc.java
     |...
     |947. try {
```

[Table of Contents] [Description]

Path 1094:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
978. public int getNumberOfPosts_inThread(int threadID, boolean
enable) //postdaoimpljdbc.java
     |
```

```
...
993. try {
```

[Table of Contents] [Description]

Path 1095:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1012. public void updateParentPostID(int oldParentPostID, int
newParentPostID) //postdaoimpljdbc.java
...
1020. try {
```

[Table of Contents] [Description]

Path 1096:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1046. public void increaseEditCount(int postID) //postdaoimpljdbc.java
...
1052. try {
```

[Table of Contents] [Description]

Path 1097:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1078. public Collection getLastEnablePosts_inThread_limit(int threadID, int
rowsToReturn) //postdaoimpljdbc.java
...
1091. try {
```

[Table of Contents] [Description]

Path 1098:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1078. public Collection getLastEnablePosts_inThread_limit(int threadID, int
rowsToReturn) //postdaoimpljdbc.java
...
1095. try {
```

[Table of Contents] [Description]

Path 1099:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1146. public Collection getLastEnablePosts_inForum_limit(int forumID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
1159. try {
```

[Table of Contents] [Description]

Path 1100:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1146. public Collection getLastEnablePosts_inForum_limit(int forumID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
1163. try {
```

[Table of Contents] [Description]

Path 1101:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1207. public int getNumberOfPosts() //postdaoimpljdbc.java
      ...
1217. try {
```

[Table of Contents] [Description]

Path 1102:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1239. public Collection getPosts() //postdaoimpljdbc.java
      ...
1251. try {
```

[Table of Contents] [Description]

Path 1103:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1280. public int getMaxPostID() //postdaoimpljdbc.java
      ...
1289. try {
```

[Table of Contents] [Description]

Path 1104:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1312. public Collection getPosts_fromIDRange(int fromID, int
toID) //postdaoimpljdbc.java
      |...
      |1327. try {
```

[Table of Contents] [Description]

Path 1105:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1373. public Collection getMostActiveMembers(Timestamp since, int
rowsToReturn) //postdaoimpljdbc.java
      |...
      |1391. try {
```

[Table of Contents] [Description]

Path 1106:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1416. public Collection getMostActiveThreads(Timestamp since, int
rowsToReturn) //postdaoimpljdbc.java
      |...
      |1437. try {
```

[Table of Contents] [Description]

Path 1107:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public void findByAlternateKey_RankTitle(String
rankTitle) //rankdaoimpljdbc.java
      |...
      |86. try {
```

[Table of Contents] [Description]

Path 1108:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public void findByAlternateKey_RankMinPosts(int
rankMinPosts) //rankdaoimpljdbc.java
      |
```

```
...
114. try {
```

[Table of Contents] [Description]

Path 1109:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
...
143. try {
```

[Table of Contents] [Description]

Path 1110:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
...
154. try {
```

[Table of Contents] [Description]

Path 1111:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
...
168. try {
```

[Table of Contents] [Description]

Path 1112:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
197. public void update(int rankID, // primary key //rankdaoimpljdbc.java
...
206. try {
```

[Table of Contents] [Description]

Path 1113:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
197. public void update(int rankID, // primary key //rankdaoimpljdbc.java
       ...
216. try {
```

[Table of Contents] [Description]

Path 1114:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
197. public void update(int rankID, // primary key //rankdaoimpljdbc.java
       ...
229. try {
```

[Table of Contents] [Description]

Path 1115:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
257. public void delete(int rankID) //rankdaoimpljdbc.java
       ...
266. try {
```

[Table of Contents] [Description]

Path 1116:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
288. public RankBean getRank(int rankID) //rankdaoimpljdbc.java
       ...
298. try {
```

[Table of Contents] [Description]

Path 1117:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
332. public Collection getRanks() //rankdaoimpljdbc.java
       ...
344. try {
```

[Table of Contents] [Description]

Path 1118:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
370. public int getRankIDFromRankTitle(String rankTitle) //rankdaoimpljdbc.java
       ...
378. try {
```

[Table of Contents] [Description]

Path 1119:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. public void findByPrimaryKey(int threadID) //threaddaoimpljdbc.java
       ...
84. try {
```

[Table of Contents] [Description]

Path 1120:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
       ...
119. try {
```

[Table of Contents] [Description]

Path 1121:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
       ...
130. try {
```

[Table of Contents] [Description]

Path 1122:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
       ...
143. try {
```

[Table of Contents] [Description]

Path 1123:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
      ...
157. try {
```

[Table of Contents] [Description]

Path 1124:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
198. public int createThread(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
      ...
208. try {
```

[Table of Contents] [Description]

Path 1125:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
218. public void delete(int threadID) //threaddaoimpljdbc.java
      ...
227. try {
```

[Table of Contents] [Description]

Path 1126:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
244. public void delete_inForum(int forumID) //threaddaoimpljdbc.java
      ...
253. try {
```

[Table of Contents] [Description]

Path 1127:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
276. public void updateTopic_Body_Icon(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
285. try {
```

[Table of Contents] [Description]

Path 1128:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
316. public void updateThreadAttachCount(int threadID, //threaddaoimpljdbc.java
       ...
325. try {
```

[Table of Contents] [Description]

Path 1129:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
351. public void increaseReplyCount(int threadID) //threaddaoimpljdbc.java
       ...
357. try {
```

[Table of Contents] [Description]

Path 1130:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
383. public void updateLastPostMemberName(int threadID, // primary
key //threaddaoimpljdbc.java
       ...
389. try {
```

[Table of Contents] [Description]

Path 1131:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
383. public void updateLastPostMemberName(int threadID, // primary
key //threaddaoimpljdbc.java
       ...
405. try {
```

[Table of Contents] [Description]

Path 1132:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
441. public void updateLastPostDate(int threadID, // primary
key //threaddaoimpljdbc.java
```

```
...
450. try {
```

[Table of Contents] [Description]

Path 1133:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
480. public ThreadBean getThread(int threadID) //threaddaoimpljdbc.java
...
490. try {
```

[Table of Contents] [Description]

Path 1134:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
546. private int getNumberOfBeans_inForum(int forumID, boolean
enable) //threaddaoimpljdbc.java
...
561. try {
```

[Table of Contents] [Description]

Path 1135:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
578. public int getNumberOfNormalEnableThreads_inForum(int
forumID) //threaddaoimpljdbc.java
...
589. try {
```

[Table of Contents] [Description]

Path 1136:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
618. private int getNumberOfBeans(boolean enable, boolean onlyNoReply, boolean
checkForumStatus, int status, int category, int forum) //threaddaoimpljdbc.java
...
654. try {
```

[Table of Contents] [Description]

Path 1137:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
696. private int findThreadID(int forumID, String memberName, Timestamp
threadCreationDate) //threaddaoimpljdbc.java
     ...
706. try {
```

[Table of Contents] [Description]

Path 1138:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
735. public Collection getThreads_inFavorite_inMember(int
memberID) //threaddaoimpljdbc.java
     ...
747. try {
```

[Table of Contents] [Description]

Path 1139:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
845. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
     802. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
     rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 1140:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
959. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
     ...
976. try {
```

[Table of Contents] [Description]

Path 1141:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1018. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
```

```
forum) //threaddaoimpljdbc.java
      |...
      |1072. try {
```

[Table of Contents] [Description]

Path 1142:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1132. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
      |...
      |1148. try {
```

[Table of Contents] [Description]

Path 1143:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
      |...
      |1248. try {
```

[Table of Contents] [Description]

Path 1144:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
      |...
      |1268. try {
```

[Table of Contents] [Description]

Path 1145:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1313. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
      |
```

```
...
1328. try {
```

[Table of Contents] [Description]

Path 1146:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1313. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
...
1334. try {
```

[Table of Contents] [Description]

Path 1147:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1414. private Collection getBeans_inForum_withSortSupport_limit_mysql(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1452. try {
```

[Table of Contents] [Description]

Path 1148:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1504. private Collection getBeans_inForum_withSortSupport_limit_noscroll(int
forumID, int offset, int rowsToReturn, String sort, String order, boolean enable,
boolean onlyNormalThread) //threaddaoimpljdbc.java
...
1541. try {
```

[Table of Contents] [Description]

Path 1149:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1634. try {
```

[Table of Contents] [Description]

Path 1150:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
      ...
1639. try {
```

[Table of Contents] [Description]

Path 1151:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1687. public void updateForumID(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
1691. try {
```

[Table of Contents] [Description]

Path 1152:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1687. public void updateForumID(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
1704. try {
```

[Table of Contents] [Description]

Path 1153:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1730. public void updateThreadStatus(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
1741. try {
```

[Table of Contents] [Description]

Path 1154:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1767. public void updateThreadType(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
1778. try {
```

[Table of Contents] [Description]

Path 1155:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1804. public void increaseViewCount(int threadID) //threaddaoimpljdbc.java
      ...
1810. try {
```

[Table of Contents] [Description]

Path 1156:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1829. public void updateReplyCount(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
1842. try {
```

[Table of Contents] [Description]

Path 1157:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1865. public void updateThreadHasPoll(int threadID, int
pollCount) //threaddaoimpljdbc.java
      ...
1873. try {
```

[Table of Contents] [Description]

Path 1158:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1893. public int getPreviousEnableThread(int forumID, int
threadID) //threaddaoimpljdbc.java
      ...
1900. try {
```

[Table of Contents] [Description]

Path 1159:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1918. public int getNextEnableThread(int forumID, int
threadID) //threaddaoimpljdbc.java
      ...
1925. try {
```

[Table of Contents] [Description]

Path 1160:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1950. public Collection getEnableThreads_inGlobal(Timestamp
sinceDate) //threaddaoimpljdbc.java
      ...
1962. try {
```

[Table of Contents] [Description]

Path 1161:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2008. public Collection getEnableThreads_inCategory(int categoryID, Timestamp
sinceDate) //threaddaoimpljdbc.java
      ...
2020. try {
```

[Table of Contents] [Description]

Path 1162:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2067. public Collection getEnableThreads_inForum(int forumID, Timestamp
sinceDate) //threaddaoimpljdbc.java
      ...
2079. try {
```

[Table of Contents] [Description]

Path 1163:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2126. public Collection getEnableThreads_inThread(int threadID, Timestamp
sinceDate) //threaddaoimpljdbc.java
      |
```

```
...
2138. try {
```

[Table of Contents] [Description]

Path 1164:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2178. public int getNumberOfEnableThreadsWithPendingPosts() //threaddaoimpljdbc.java
...
2188. try {
```

[Table of Contents] [Description]

Path 1165:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2204. public int getNumberOfEnableThreadsWithPendingPosts_inForum(int
forumID) //threaddaoimpljdbc.java
...
2214. try {
```

[Table of Contents] [Description]

Path 1166:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2275. private Collection
getEnableThreadsWithPendingPosts_withSortSupport_limit_noscroll(int forumID, int
offset, int rowsToReturn, String sort, String order) //threaddaoimpljdbc.java
...
2315. try {
```

[Table of Contents] [Description]

Path 1167:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
2359. private Collection getEnableThreads_inType_inForum(int forumID, int
threadType, boolean checkForumStatus) //threaddaoimpljdbc.java
...
2381. try {
```

[Table of Contents] [Description]

Path 1168:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public void findByPrimaryKey(int watchID) //watchdaoimpljdbc.java
      ...
      83. try {
```

[Table of Contents] [Description]

Path 1169:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
101. public void findByAlternateKey_MemberID_CategoryID_ForumID_ThreadID(int
memberID, int categoryID, int forumID, int threadID) //watchdaoimpljdbc.java
      ...
      111. try {
```

[Table of Contents] [Description]

Path 1170:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
      ...
      162. try {
```

[Table of Contents] [Description]

Path 1171:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
      ...
      171. try {
```

[Table of Contents] [Description]

Path 1172:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
      ...
      179. try {
```

[Table of Contents] [Description]

Path 1173:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
     |...
     189. try {
```

[Table of Contents] [Description]

Path 1174:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
     |...
     199. try {
```

[Table of Contents] [Description]

Path 1175:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
     |...
     214. try {
```

[Table of Contents] [Description]

Path 1176:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
242. public void delete(int watchID) //watchdaoimpljdbc.java
     |...
     251. try {
```

[Table of Contents] [Description]

Path 1177:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
268. public void delete_inMember(int memberID) //watchdaoimpljdbc.java
```

```
...
277. try {
```

[Table of Contents] [Description]

Path 1178:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
292. public void delete_inCategory(int categoryID) //watchdaoimpljdbc.java
...
301. try {
```

[Table of Contents] [Description]

Path 1179:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
316. public void delete_inForum(int forumID) //watchdaoimpljdbc.java
...
325. try {
```

[Table of Contents] [Description]

Path 1180:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
340. public void delete_inThread(int threadID) //watchdaoimpljdbc.java
...
349. try {
```

[Table of Contents] [Description]

Path 1181:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
369. public void updateLastSentDate(int watchID, // primary
key //watchdaoimpljdbc.java
...
378. try {
```

[Table of Contents] [Description]

Path 1182:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
406. public WatchBean getWatch(int watchID) //watchdaoimpljdbc.java
       ...
416. try {
```

[Table of Contents] [Description]

Path 1183:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
454. public WatchBean getWatch_byAlternateKey_MemberID_CategoryID_ForumID_ThreadID
(int memberID, int categoryID, int forumID, int threadID) //watchdaoimpljdbc.java
       ...
464. try {
```

[Table of Contents] [Description]

Path 1184:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
506. public Collection getWatches() //watchdaoimpljdbc.java
       ...
518. try {
```

[Table of Contents] [Description]

Path 1185:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
548. public int getNumberOfWatches() //watchdaoimpljdbc.java
       ...
558. try {
```

[Table of Contents] [Description]

Path 1186:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
574. public int getNumberOfWatches_forMember(int memberID) //watchdaoimpljdbc.java
       ...
584. try {
```

[Table of Contents] [Description]

Path 1187:

Query Name - **Catch_Without_General_Exception_Handling**

Severity

- **Information**

```
610. public Collection getMemberBeans() //watchdaoimpljdbc.java
       ...
623. try {
```

[Table of Contents] [Description]

Path 1188:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
650. public Collection getWatches_forMember(int memberID) //watchdaoimpljdbc.java
       ...
662. try {
```

[Table of Contents] [Description]

Path 1189:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
698. public void updateLastSentDate_forMember(int memberID, //watchdaoimpljdbc.java
       ...
707. try {
```

[Table of Contents] [Description]

Path 1190:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
730. public void updateWatchType(int watchID, int watchType) //watchdaoimpljdbc.java
       ...
738. try {
```

[Table of Contents] [Description]

Path 1191:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //attachmentindexer.java
       ...
150. try {
```

[Table of Contents] [Description]

Path 1192:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
143. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //attachmentindexer.java
     |...
     |163. try {
```

[Table of Contents] [Description]

Path 1193:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
185. static void doIndexAttachment(AttachmentBean attachmentBean, IndexWriter
writer) throws SearchException //attachmentindexer.java
     |...
     |204. try {
```

[Table of Contents] [Description]

Path 1194:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addAttachmentToIndex(AttachmentBean attachmentBean) throws
SearchException, IOException //attachmentindexer.java
     |...
     |227. try {
```

[Table of Contents] [Description]

Path 1195:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addAttachmentToIndex(AttachmentBean attachmentBean) throws
SearchException, IOException //attachmentindexer.java
     |...
     |249. try {
```

[Table of Contents] [Description]

Path 1196:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addAttachmentToIndex(AttachmentBean attachmentBean) throws
SearchException, IOException //attachmentindexer.java
     |...
     |256. try {
```

[Table of Contents] [Description]

Path 1197:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
270. static void deleteAttachmentFromIndex(int attachmentID) throws
SearchException //attachmentindexer.java
     ...
274. try {
```

[Table of Contents] [Description]

Path 1198:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
270. static void deleteAttachmentFromIndex(int attachmentID) throws
SearchException //attachmentindexer.java
     ...
291. try {
```

[Table of Contents] [Description]

Path 1199:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
270. static void deleteAttachmentFromIndex(int attachmentID) throws
SearchException //attachmentindexer.java
     ...
298. try {
```

[Table of Contents] [Description]

Path 1200:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
307. public static int getNumDocs() //attachmentindexer.java
     ...
312. try {
```

[Table of Contents] [Description]

Path 1201:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
307. public static int getNumDocs() //attachmentindexer.java
     ...
```

```
326. try {
```

[Table of Contents] [Description]

Path 1202:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
307. public static int getNumDocs() //attachmentindexer.java
     ...
333. try {
```

[Table of Contents] [Description]

Path 1203:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. protected IndexSearcher getSearcher(Directory directory) throws
IOException //attachmentsearchquery.java
130. try {
```

[Table of Contents] [Description]

Path 1204:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //attachmentsearchquery.java
     ...
154. try {
```

[Table of Contents] [Description]

Path 1205:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //attachmentsearchquery.java
     ...
195. try {
```

[Table of Contents] [Description]

Path 1206:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
142. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //attachmentsearchquery.java
      ...
218. try {
```

[Table of Contents] [Description]

Path 1207:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void run() //rebuildattachmentindextask.java
      ...
94. try {
```

[Table of Contents] [Description]

Path 1208:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void run() //rebuildattachmentindextask.java
      ...
129. try {
```

[Table of Contents] [Description]

Path 1209:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
88. public void run() //rebuildattachmentindextask.java
      ...
136. try {
```

[Table of Contents] [Description]

Path 1210:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //memberindexer.java
      ...
151. try {
```

[Table of Contents] [Description]

Path 1211:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //memberindexer.java
     |...
     |164. try {
```

[Table of Contents] [Description]

Path 1212:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
186. static void doIndexMember(MemberBean memberBean, IndexWriter writer) throws
SearchException //memberindexer.java
     |...
     |204. try {
```

[Table of Contents] [Description]

Path 1213:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addMemberToIndex(MemberBean memberBean) throws SearchException,
IOException //memberindexer.java
     |...
     |227. try {
```

[Table of Contents] [Description]

Path 1214:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addMemberToIndex(MemberBean memberBean) throws SearchException,
IOException //memberindexer.java
     |...
     |248. try {
```

[Table of Contents] [Description]

Path 1215:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
222. static void addMemberToIndex(MemberBean memberBean) throws SearchException,
IOException //memberindexer.java
     |...
     |255. try {
```

[Table of Contents] [Description]

Path 1216:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. static void deleteMemberFromIndex(int memberID) throws
SearchException //memberindexer.java
     ...
273. try {
```

[Table of Contents] [Description]

Path 1217:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. static void deleteMemberFromIndex(int memberID) throws
SearchException //memberindexer.java
     ...
290. try {
```

[Table of Contents] [Description]

Path 1218:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. static void deleteMemberFromIndex(int memberID) throws
SearchException //memberindexer.java
     ...
297. try {
```

[Table of Contents] [Description]

Path 1219:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
306. public static int getNumDocs() //memberindexer.java
     ...
311. try {
```

[Table of Contents] [Description]

Path 1220:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
306. public static int getNumDocs() //memberindexer.java
     ...
325. try {
```

[Table of Contents] [Description]

Path 1221:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
306. public static int getNumDocs() //memberindexer.java
      |...
      |332. try {
```

[Table of Contents] [Description]

Path 1222:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
141. protected IndexSearcher getSearcher(Directory directory) throws
IOException //membersearchquery.java
      |142. try {
```

[Table of Contents] [Description]

Path 1223:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. public void searchDocuments(int offset, int
rowsToReturn) //membersearchquery.java
      |...
      |160. try {
```

[Table of Contents] [Description]

Path 1224:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. public void searchDocuments(int offset, int
rowsToReturn) //membersearchquery.java
      |...
      |207. try {
```

[Table of Contents] [Description]

Path 1225:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. public void searchDocuments(int offset, int
rowsToReturn) //membersearchquery.java
```

```
...
228. try {
```

[Table of Contents] [Description]

Path 1226:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildmemberindextask.java
...
93. try {
```

[Table of Contents] [Description]

Path 1227:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildmemberindextask.java
...
131. try {
```

[Table of Contents] [Description]

Path 1228:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildmemberindextask.java
...
138. try {
```

[Table of Contents] [Description]

Path 1229:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //postindexer.java
...
161. try {
```

[Table of Contents] [Description]

Path 1230:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
154. static IndexWriter getIndexWriter(Directory directory, boolean create) throws
SearchException //postindexer.java
    ...
175. try {
```

[Table of Contents] [Description]

Path 1231:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
198. static void doIndexPost(PostBean post, IndexWriter writer) throws
SearchException //postindexer.java
    ...
225. try {
```

[Table of Contents] [Description]

Path 1232:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
243. static void addPostToIndex(PostBean post) throws SearchException,
IOException //postindexer.java
    ...
248. try {
```

[Table of Contents] [Description]

Path 1233:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
243. static void addPostToIndex(PostBean post) throws SearchException,
IOException //postindexer.java
    ...
269. try {
```

[Table of Contents] [Description]

Path 1234:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
243. static void addPostToIndex(PostBean post) throws SearchException,
IOException //postindexer.java
    ...
276. try {
```

[Table of Contents] [Description]

Path 1235:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
290. static void deletePostFromIndex(int postID) throws
SearchException //postindexer.java
     |...
     |294. try {
```

[Table of Contents] [Description]

Path 1236:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
290. static void deletePostFromIndex(int postID) throws
SearchException //postindexer.java
     |...
     |311. try {
```

[Table of Contents] [Description]

Path 1237:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
290. static void deletePostFromIndex(int postID) throws
SearchException //postindexer.java
     |...
     |318. try {
```

[Table of Contents] [Description]

Path 1238:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
332. static void deleteThreadFromIndex(int threadID) throws
SearchException //postindexer.java
     |...
     |336. try {
```

[Table of Contents] [Description]

Path 1239:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
332. static void deleteThreadFromIndex(int threadID) throws
SearchException //postindexer.java
     |...
```

```
353. try {
```

[Table of Contents] [Description]

Path 1240:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
332. static void deleteThreadFromIndex(int threadID) throws
SearchException //postindexer.java
...
360. try {
```

[Table of Contents] [Description]

Path 1241:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
374. static void deleteForumFromIndex(int forumID) throws
SearchException //postindexer.java
...
378. try {
```

[Table of Contents] [Description]

Path 1242:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
374. static void deleteForumFromIndex(int forumID) throws
SearchException //postindexer.java
...
395. try {
```

[Table of Contents] [Description]

Path 1243:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
374. static void deleteForumFromIndex(int forumID) throws
SearchException //postindexer.java
...
402. try {
```

[Table of Contents] [Description]

Path 1244:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
411. public static int getNumDocs() //postindexer.java
     |...
     |416. try {
```

[Table of Contents] [Description]

Path 1245:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
411. public static int getNumDocs() //postindexer.java
     |...
     |430. try {
```

[Table of Contents] [Description]

Path 1246:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
411. public static int getNumDocs() //postindexer.java
     |...
     |437. try {
```

[Table of Contents] [Description]

Path 1247:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
165. protected IndexSearcher getSearcher(Directory directory) throws
IOException //postsearchquery.java
     |166. try {
```

[Table of Contents] [Description]

Path 1248:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
178. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //postsearchquery.java
     |...
     |190. try {
```

[Table of Contents] [Description]

Path 1249:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
178. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //postsearchquery.java
     ...
250. try {
```

[Table of Contents] [Description]

Path 1250:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
178. public void searchDocuments(int offset, int rowsToReturn, MVNForumPermission
permission) //postsearchquery.java
     ...
270. try {
```

[Table of Contents] [Description]

Path 1251:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildpostindextask.java
     ...
93. try {
```

[Table of Contents] [Description]

Path 1252:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildpostindextask.java
     ...
128. try {
```

[Table of Contents] [Description]

Path 1253:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
87. public void run() //rebuildpostindextask.java
     ...
135. try {
```

[Table of Contents] [Description]

Path 1254:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
      ...
90. try {
```

[Table of Contents] [Description]

Path 1255:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
      ...
119. try {
```

[Table of Contents] [Description]

Path 1256:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
      ...
126. try {
```

[Table of Contents] [Description]

Path 1257:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
      ...
85. try {
```

[Table of Contents] [Description]

Path 1258:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
      ...
94. try {
```

[Table of Contents] [Description]

Path 1259:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
    ...
106. try {
```

[Table of Contents] [Description]

Path 1260:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
    ...
112. try {
```

[Table of Contents] [Description]

Path 1261:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
    ...
121. try {
```

[Table of Contents] [Description]

Path 1262:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
    ...
130. try {
```

[Table of Contents] [Description]

Path 1263:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
```

```
...
142. try {
```

[Table of Contents] [Description]

Path 1264:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
...
148. try {
```

[Table of Contents] [Description]

Path 1265:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
...
157. try {
```

[Table of Contents] [Description]

Path 1266:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
...
165. try {
```

[Table of Contents] [Description]

Path 1267:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
...
195. try {
```

[Table of Contents] [Description]

Path 1268:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public BinaryStorageHandle storeData(String category, String nameId, String
fileName, InputStream inputStream, //binarystorageserviceimpldisk.java
    ...
201. try {
```

[Table of Contents] [Description]

Path 1269:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public InputStream getInputStream(String category, String nameId,
BinaryStorageHandle handle) //binarystorageserviceimpldisk.java
    ...
218. try {
```

[Table of Contents] [Description]

Path 1270:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public InputStream getInputStream(String category, String nameId,
BinaryStorageHandle handle) //binarystorageserviceimpldisk.java
    ...
239. try {
```

[Table of Contents] [Description]

Path 1271:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public InputStream getInputStream(String category, String nameId,
BinaryStorageHandle handle) //binarystorageserviceimpldisk.java
    ...
260. try {
```

[Table of Contents] [Description]

Path 1272:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
213. public InputStream getInputStream(String category, String nameId,
BinaryStorageHandle handle) //binarystorageserviceimpldisk.java
    ...
```

```
267. try {
```

[Table of Contents] [Description]

Path 1273:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
     ...
305. try {
```

[Table of Contents] [Description]

Path 1274:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
     ...
322. try {
```

[Table of Contents] [Description]

Path 1275:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
     ...
339. try {
```

[Table of Contents] [Description]

Path 1276:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
83. public void contextInitialized(ServletContextEvent
event) //mvnforumlifecycleserviceimpldefault.java
     ...
103. try {
```

[Table of Contents] [Description]

Path 1277:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
96. public MvnForumAdService getMvnForumAdService
() //mvnforumserviceimpldefault.java
    ...
99. try {
```

[Table of Contents] [Description]

Path 1278:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
56. public ActionInUserModule(GenericRequest request, String requestURI) throws
MissingURLMapEntryException //actioninusermodule.java
    ...
82. try {
```

[Table of Contents] [Description]

Path 1279:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
56. public ActionInUserModule(GenericRequest request, String requestURI) throws
MissingURLMapEntryException //actioninusermodule.java
    ...
97. try {
```

[Table of Contents] [Description]

Path 1280:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
56. public ActionInUserModule(GenericRequest request, String requestURI) throws
MissingURLMapEntryException //actioninusermodule.java
    ...
135. try {
```

[Table of Contents] [Description]

Path 1281:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
56. public ActionInUserModule(GenericRequest request, String requestURI) throws
MissingURLMapEntryException //actioninusermodule.java
    ...
147. try {
```

[Table of Contents] [Description]

Path 1282:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
56. public ActionInUserModule(GenericRequest request, String requestURI) throws
MissingURLMapEntryException //actioninusermodule.java
    ...
157. try {
```

[Table of Contents] [Description]

Path 1283:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public void prepareAdd(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
    ...
118. try {
```

[Table of Contents] [Description]

Path 1284:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
146. public void processAdd(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
    ...
176. try {
```

[Table of Contents] [Description]

Path 1285:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
146. public void processAdd(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
    ...
251. try {
```

[Table of Contents] [Description]

Path 1286:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
348. public void prepareEdit(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
```

```
...
359. try {
```

[Table of Contents] [Description]

Path 1287:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
348. public void prepareEdit(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
...
368. try {
```

[Table of Contents] [Description]

Path 1288:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
397. public void processEdit(GenericRequest request) //attachmentwebhandler.java
...
415. try {
```

[Table of Contents] [Description]

Path 1289:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
461. public void prepareDelete(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
...
472. try {
```

[Table of Contents] [Description]

Path 1290:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
461. public void prepareDelete(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
...
481. try {
```

[Table of Contents] [Description]

Path 1291:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
511. public void processDelete(GenericRequest request) //attachmentwebhandler.java
      ...
529. try {
```

[Table of Contents] [Description]

Path 1292:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
511. public void processDelete(GenericRequest request) //attachmentwebhandler.java
      ...
547. try {
```

[Table of Contents] [Description]

Path 1293:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
      ...
607. try {
```

[Table of Contents] [Description]

Path 1294:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
      ...
649. try {
```

[Table of Contents] [Description]

Path 1295:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
      ...
664. try {
```

[Table of Contents] [Description]

Path 1296:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
     ...
685. try {
```

[Table of Contents] [Description]

Path 1297:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
     ...
690. try {
```

[Table of Contents] [Description]

Path 1298:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
702. static void deleteAttachments_inPost(int postID) throws
DatabaseException //attachmentwebhandler.java
     ...
715. try {
```

[Table of Contents] [Description]

Path 1299:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
729. static void deleteAttachments_inThread(int threadID) throws
DatabaseException //attachmentwebhandler.java
     ...
741. try {
```

[Table of Contents] [Description]

Path 1300:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
755. public void prepareListAttachments(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
```

```
...
765. try {
```

[Table of Contents] [Description]

Path 1301:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
...
91. try {
```

[Table of Contents] [Description]

Path 1302:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
...
105. try {
```

[Table of Contents] [Description]

Path 1303:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
101. public void process(HttpServletRequest request, HttpServletResponse
response) //forumuserservlet.java
...
116. try {
```

[Table of Contents] [Description]

Path 1304:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
101. public void process(HttpServletRequest request, HttpServletResponse
response) //forumuserservlet.java
...
120. try {
```

[Table of Contents] [Description]

Path 1305:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
66. public void prepareSendMail(GenericRequest request) //mailwebhandler.java
      ...
93. try {
```

[Table of Contents] [Description]

Path 1306:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
      ...
142. try {
```

[Table of Contents] [Description]

Path 1307:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void processAdd(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
155. try {
```

[Table of Contents] [Description]

Path 1308:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void processAdd(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
308. try {
```

[Table of Contents] [Description]

Path 1309:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
129. public void processAdd(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
336. try {
```

[Table of Contents] [Description]

Path 1310:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
360. public void processUpdate(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
504. try {
```

[Table of Contents] [Description]

Path 1311:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
735. public void updateMemberAvatar(GenericRequest request) //memberwebhandler.java
      ...
765. try {
```

[Table of Contents] [Description]

Path 1312:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
782. public void uploadAvatar(javax.servlet.ServletConfig config, GenericRequest
request) //memberwebhandler.java
      ...
805. try {
```

[Table of Contents] [Description]

Path 1313:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
900. try {
```

[Table of Contents] [Description]

Path 1314:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
985. try {
```

[Table of Contents] [Description]

Path 1315:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
      ...
1017. try {
```

[Table of Contents] [Description]

Path 1316:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
      ...
1064. try {
```

[Table of Contents] [Description]

Path 1317:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1084. public void activateMember(GenericRequest request) //memberwebhandler.java
      ...
1103. try {
```

[Table of Contents] [Description]

Path 1318:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1150. try {
```

[Table of Contents] [Description]

Path 1319:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1158. try {
```

[Table of Contents] [Description]

Path 1320:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1174. public void prepareListMembers_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1204. try {
```

[Table of Contents] [Description]

Path 1321:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1224. public void getAvatar(HttpServletRequest request, HttpServletResponse
response) //memberwebhandler.java
      ...
1235. try {
```

[Table of Contents] [Description]

Path 1322:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1224. public void getAvatar(HttpServletRequest request, HttpServletResponse
response) //memberwebhandler.java
      ...
1277. try {
```

[Table of Contents] [Description]

Path 1323:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1224. public void getAvatar(HttpServletRequest request, HttpServletResponse
response) //memberwebhandler.java
      ...
1286. try {
```

[Table of Contents] [Description]

Path 1324:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1224. public void getAvatar(HttpServletRequest request, HttpServletResponse
response) //memberwebhandler.java
      ...
1305. try {
```

[Table of Contents] [Description]

Path 1325:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
155. public void prepareDelete(GenericRequest
request) //messagefolderwebhandler.java
      ...
184. try {
```

[Table of Contents] [Description]

Path 1326:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java
      ...
233. try {
```

[Table of Contents] [Description]

Path 1327:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
      ...
265. try {
```

[Table of Contents] [Description]

Path 1328:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
70. public void prepareAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
102. try {
```

[Table of Contents] [Description]

Path 1329:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
70. public void prepareAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
148. try {
```

[Table of Contents] [Description]

Path 1330:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
190. try {
```

[Table of Contents] [Description]

Path 1331:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
278. try {
```

[Table of Contents] [Description]

Path 1332:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
324. try {
```

[Table of Contents] [Description]

Path 1333:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
```

```
response) //messagewebhandler.java
    ...
358. try {
```

[Table of Contents] [Description]

Path 1334:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
    ...
403. try {
```

[Table of Contents] [Description]

Path 1335:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
    ...
430. try {
```

[Table of Contents] [Description]

Path 1336:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
    ...
473. try {
```

[Table of Contents] [Description]

Path 1337:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
    ...
490. try {
```

[Table of Contents] [Description]

Path 1338:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
514. public void prepareList(GenericRequest request) //messagewebhandler.java
     ...
550. try {
```

[Table of Contents] [Description]

Path 1339:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
514. public void prepareList(GenericRequest request) //messagewebhandler.java
     ...
557. try {
```

[Table of Contents] [Description]

Path 1340:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
     ...
669. try {
```

[Table of Contents] [Description]

Path 1341:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
     ...
698. try {
```

[Table of Contents] [Description]

Path 1342:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
     ...
725. try {
```

[Table of Contents] [Description]

Path 1343:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
746. public void processSendMessage(GenericRequest request) //messagewebhandler.java
     ...
767. try {
```

[Table of Contents] [Description]

Path 1344:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
746. public void processSendMessage(GenericRequest request) //messagewebhandler.java
     ...
832. try {
```

[Table of Contents] [Description]

Path 1345:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
850. public void processDelete(GenericRequest request) //messagewebhandler.java
     ...
872. try {
```

[Table of Contents] [Description]

Path 1346:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
83. public void prepareAdd(GenericRequest request) //pmattachmentwebhandler.java
    ...
103. try {
```

[Table of Contents] [Description]

Path 1347:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
154. try {
```

[Table of Contents] [Description]

Path 1348:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
235. try {
```

[Table of Contents] [Description]

Path 1349:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
274. try {
```

[Table of Contents] [Description]

Path 1350:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
300. try {
```

[Table of Contents] [Description]

Path 1351:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
351. try {
```

[Table of Contents] [Description]

Path 1352:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
     ...
```

```
374. try {
```

[Table of Contents] [Description]

Path 1353:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
417. try {
```

[Table of Contents] [Description]

Path 1354:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
425. try {
```

[Table of Contents] [Description]

Path 1355:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
446. try {
```

[Table of Contents] [Description]

Path 1356:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
458. try {
```

[Table of Contents] [Description]

Path 1357:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
477. try {
```

[Table of Contents] [Description]

Path 1358:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
494. try {
```

[Table of Contents] [Description]

Path 1359:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
499. try {
```

[Table of Contents] [Description]

Path 1360:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
506. public void deleteOrphanPmAttachment() throws
DatabaseException //pmattachmentwebhandler.java
     ...
526. try {
```

[Table of Contents] [Description]

Path 1361:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
     ...
86. try {
```

[Table of Contents] [Description]

Path 1362:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
      ...
98. try {
```

[Table of Contents] [Description]

Path 1363:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
      ...
106. try {
```

[Table of Contents] [Description]

Path 1364:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
      ...
121. try {
```

[Table of Contents] [Description]

Path 1365:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
      ...
138. try {
```

[Table of Contents] [Description]

Path 1366:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public static void addPost(int parentPostID, int forumID, int memberID, String
currentIP, String memberName, //postutil.java
      ...
171. try {
```

[Table of Contents] [Description]

Path 1367:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
181. public static void sendEmailToAdminBecauseCensoredPost(GenericRequest request,
int postID, int forumID, int threadID, int memberID) //postutil.java
      ...
211. try {
```

[Table of Contents] [Description]

Path 1368:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
129. try {
```

[Table of Contents] [Description]

Path 1369:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
151. try {
```

[Table of Contents] [Description]

Path 1370:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
162. try {
```

[Table of Contents] [Description]

Path 1371:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
179. try {
```

[Table of Contents] [Description]

Path 1372:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
256. try {
```

[Table of Contents] [Description]

Path 1373:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
267. try {
```

[Table of Contents] [Description]

Path 1374:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
306. try {
```

[Table of Contents] [Description]

Path 1375:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
```

```
response) //postwebhandler.java
    ...
358. try {
```

[Table of Contents] [Description]

Path 1376:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
368. try {
```

[Table of Contents] [Description]

Path 1377:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
384. try {
```

[Table of Contents] [Description]

Path 1378:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
401. try {
```

[Table of Contents] [Description]

Path 1379:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
453. try {
```

[Table of Contents] [Description]

Path 1380:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
     ...
481. try {
```

[Table of Contents] [Description]

Path 1381:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
     ...
500. try {
```

[Table of Contents] [Description]

Path 1382:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
535. public void addPostSuccessForRender(GenericRequest request, GenericResponse
response) //postwebhandler.java
     ...
545. try {
```

[Table of Contents] [Description]

Path 1383:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
     ...
573. try {
```

[Table of Contents] [Description]

Path 1384:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
     ...
582. try {
```

[Table of Contents] [Description]

Path 1385:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
       ...
593. try {
```

[Table of Contents] [Description]

Path 1386:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
637. public void prepareEdit(GenericRequest request, GenericResponse
response) //postwebhandler.java
       ...
651. try {
```

[Table of Contents] [Description]

Path 1387:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
637. public void prepareEdit(GenericRequest request, GenericResponse
response) //postwebhandler.java
       ...
661. try {
```

[Table of Contents] [Description]

Path 1388:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
637. public void prepareEdit(GenericRequest request, GenericResponse
response) //postwebhandler.java
       ...
673. try {
```

[Table of Contents] [Description]

Path 1389:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
```

```
...
786. try {
```

[Table of Contents] [Description]

Path 1390:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
...
802. try {
```

[Table of Contents] [Description]

Path 1391:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
...
874. try {
```

[Table of Contents] [Description]

Path 1392:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
...
901. try {
```

[Table of Contents] [Description]

Path 1393:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
951. public void prepareDelete(GenericRequest request, GenericResponse
response) //postwebhandler.java
...
965. try {
```

[Table of Contents] [Description]

Path 1394:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
951. public void prepareDelete(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
973. try {
```

[Table of Contents] [Description]

Path 1395:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1013. public void processDelete(GenericRequest request) //postwebhandler.java
      ...
1023. try {
```

[Table of Contents] [Description]

Path 1396:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1054. private void deletePost(GenericRequest request, PostBean
postBean) //postwebhandler.java
      ...
1128. try {
```

[Table of Contents] [Description]

Path 1397:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1157. public void prepareModeratePendingPosts_limit(GenericRequest
request) //postwebhandler.java
      ...
1168. try {
```

[Table of Contents] [Description]

Path 1398:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1157. public void prepareModeratePendingPosts_limit(GenericRequest
request) //postwebhandler.java
      ...
1176. try {
```

[Table of Contents] [Description]

Path 1399:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
1258. try {
```

[Table of Contents] [Description]

Path 1400:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
1261. try {
```

[Table of Contents] [Description]

Path 1401:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
1285. try {
```

[Table of Contents] [Description]

Path 1402:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
1296. try {
```

[Table of Contents] [Description]

Path 1403:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
      |
```

```
...
1349. try {
```

[Table of Contents] [Description]

Path 1404:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
...
1362. try {
```

[Table of Contents] [Description]

Path 1405:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
...
1390. try {
```

[Table of Contents] [Description]

Path 1406:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
...
1409. try {
```

[Table of Contents] [Description]

Path 1407:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1487. public void processSearch(GenericRequest request, GenericResponse
response) //postwebhandler.java
...
1540. try {
```

[Table of Contents] [Description]

Path 1408:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1487. public void processSearch(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
1612. try {
```

[Table of Contents] [Description]

Path 1409:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
84. public void prepareSplit(GenericRequest request, GenericResponse
response) //threadwebhandler.java
      ...
98. try {
```

[Table of Contents] [Description]

Path 1410:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
84. public void prepareSplit(GenericRequest request, GenericResponse
response) //threadwebhandler.java
      ...
118. try {
```

[Table of Contents] [Description]

Path 1411:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
155. public void processSplit(GenericRequest request) //threadwebhandler.java
      ...
172. try {
```

[Table of Contents] [Description]

Path 1412:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
278. public void prepareDelete(GenericRequest request, GenericResponse
response) //threadwebhandler.java
      ...
290. try {
```

[Table of Contents] [Description]

Path 1413:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
278. public void prepareDelete(GenericRequest request, GenericResponse
response) //threadwebhandler.java
     ...
300. try {
```

[Table of Contents] [Description]

Path 1414:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
278. public void prepareDelete(GenericRequest request, GenericResponse
response) //threadwebhandler.java
     ...
309. try {
```

[Table of Contents] [Description]

Path 1415:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
363. public void processDelete(GenericRequest request) //threadwebhandler.java
     ...
375. try {
```

[Table of Contents] [Description]

Path 1416:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
406. public void deleteThread(GenericRequest request, ThreadBean
threadBean) //threadwebhandler.java
     ...
425. try {
```

[Table of Contents] [Description]

Path 1417:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
501. public void prepareMoveThread(GenericRequest request, GenericResponse
```

```
response) //threadwebhandler.java
    ...
514. try {
```

[Table of Contents] [Description]

Path 1418:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
552. public void processMoveThread(GenericRequest request) //threadwebhandler.java
    ...
571. try {
```

[Table of Contents] [Description]

Path 1419:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
619. public void prepareEditThreadStatus(GenericRequest request, GenericResponse
response) //threadwebhandler.java
    ...
631. try {
```

[Table of Contents] [Description]

Path 1420:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
664. public void processEditThreadStatus(GenericRequest
request) //threadwebhandler.java
    ...
682. try {
```

[Table of Contents] [Description]

Path 1421:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
721. public void prepareEditThreadType(GenericRequest request, GenericResponse
response) //threadwebhandler.java
    ...
733. try {
```

[Table of Contents] [Description]

Path 1422:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
766. public void processEditThreadType(GenericRequest
request) //threadwebhandler.java
     ...
784. try {
```

[Table of Contents] [Description]

Path 1423:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
     ...
854. try {
```

[Table of Contents] [Description]

Path 1424:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
920. public void prepareListRecentThreads_limit(GenericRequest
request,GenericResponse response) //threadwebhandler.java
     ...
937. try {
```

[Table of Contents] [Description]

Path 1425:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
973. public void prepareListRecentDisabledThreads_limit(GenericRequest
request) //threadwebhandler.java
     ...
993. try {
```

[Table of Contents] [Description]

Path 1426:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1057. public void prepareListRecentEnableThreadsWithPendingPosts_limit
(GenericRequest request) //threadwebhandler.java
     ...
```

```
1077. try {
```

[Table of Contents] [Description]

Path 1427:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1107. public void prepareListEnableThreadsWithPendingPosts_inForum_limit
(GenericRequest request) //threadwebhandler.java
...
1131. try {
```

[Table of Contents] [Description]

Path 1428:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1197. public void prepareModeratePendingThreads_inForum_limit(GenericRequest
request) //threadwebhandler.java
...
1224. try {
```

[Table of Contents] [Description]

Path 1429:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
...
1297. try {
```

[Table of Contents] [Description]

Path 1430:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
...
1318. try {
```

[Table of Contents] [Description]

Path 1431:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
      ...
1336. try {
```

[Table of Contents] [Description]

Path 1432:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1443. try {
```

[Table of Contents] [Description]

Path 1433:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1449. try {
```

[Table of Contents] [Description]

Path 1434:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1487. public void prepareListUnansweredThreads(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
1503. try {
```

[Table of Contents] [Description]

Path 1435:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public String process(HttpServletRequest request, HttpServletResponse
response) //usermoduleprocessor.java
      ...
127. try {
```

[Table of Contents] [Description]

Path 1436:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
104. public String process(HttpServletRequest request, HttpServletResponse
response) //usermoduleprocessor.java
     ...
469. try {
```

[Table of Contents] [Description]

Path 1437:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
171. public static boolean isWatched(int memberID, int categoryID, int forumID, int
threadID) throws DatabaseException //watchutil.java
     172. try {
```

[Table of Contents] [Description]

Path 1438:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
148. void sendMail_forMember(int memberID, String forumBase, Timestamp
lastSent) //watchwebhandler.java
     ...
152. try {
```

[Table of Contents] [Description]

Path 1439:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
148. void sendMail_forMember(int memberID, String forumBase, Timestamp
lastSent) //watchwebhandler.java
     ...
182. try {
```

[Table of Contents] [Description]

Path 1440:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
148. void sendMail_forMember(int memberID, String forumBase, Timestamp
lastSent) //watchwebhandler.java
     ...
```

```
259. try {
```

[Table of Contents] [Description]

Path 1441:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public void prepareList(GenericRequest request) //watchwebhandler.java
     ...
307. try {
```

[Table of Contents] [Description]

Path 1442:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
351. public void processAdd(GenericRequest request) //watchwebhandler.java
     ...
405. try {
```

[Table of Contents] [Description]

Path 1443:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1. <%-- //addmessage.jsp
     ...
59. try {
```

[Table of Contents] [Description]

Path 1444:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1. <%-- //addpost.jsp
     ...
66. try {
```

[Table of Contents] [Description]

Path 1445:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
292. public static void reload() //mvncoreconfig.java
```

```
...
370. try {
```

[Table of Contents] [Description]

Path 1446:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
204. public synchronized void load(InputStream input, String
enc) //basepropertiesconfiguration.java
...
210. try
```

[Table of Contents] [Description]

Path 1447:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
342. public Boolean getBoolean(String key, Boolean
defaultValue) //compositeconfiguration.java
...
344. try
```

[Table of Contents] [Description]

Path 1448:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
396. public Byte getByte(String key, Byte
defaultValue) //compositeconfiguration.java
...
398. try
```

[Table of Contents] [Description]

Path 1449:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
450. public Double getDouble(String key, Double
defaultValue) //compositeconfiguration.java
...
452. try
```

[Table of Contents] [Description]

Path 1450:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
488. public Float getFloat(String key, Float
defaultValue) //compositeconfiguration.java
      ...
490. try
```

[Table of Contents] [Description]

Path 1451:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
542. public Integer getInteger(String key, Integer
defaultValue) //compositeconfiguration.java
      ...
544. try
```

[Table of Contents] [Description]

Path 1452:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
596. public Long getLong(String key, Long
defaultValue) //compositeconfiguration.java
      ...
598. try
```

[Table of Contents] [Description]

Path 1453:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
650. public Short getShort(String key, Short
defaultValue) //compositeconfiguration.java
      ...
652. try
```

[Table of Contents] [Description]

Path 1454:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
682. public String getString(String key, String
defaultValue) //compositeconfiguration.java
      ...
```

```
684. try
```

[Table of Contents] [Description]

Path 1455:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
152. public Configuration getConfiguration() throws
Exception //configurationfactory.java
...
180. try
```

[Table of Contents] [Description]

Path 1456:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
620. protected Object clone() //configurationkey.java
...
622. try
```

[Table of Contents] [Description]

Path 1457:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
151. public static URL getURL(String basePath, String
file) //configurationutils.java
...
160. try
```

[Table of Contents] [Description]

Path 1458:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
269. protected void fireElementStart(String name, Attributes
attribs) //configurationxmlreader.java
...
273. try
```

[Table of Contents] [Description]

Path 1459:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
289. protected void fireElementEnd(String name) //configurationxmlreader.java
       ...
293. try
```

[Table of Contents] [Description]

Path 1460:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
308. protected void fireCharacters(String text) //configurationxmlreader.java
       ...
312. try
```

[Table of Contents] [Description]

Path 1461:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
321. private void possiblySave() //dom4jconfiguration.java
       ...
325. try
```

[Table of Contents] [Description]

Path 1462:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
893. protected Object clone() //hierarchicalconfiguration.java
       ...
895. try
```

[Table of Contents] [Description]

Path 1463:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
156. public Iterator getKeys(String key) //jndiconfiguration.java
       ...
159. try
```

[Table of Contents] [Description]

Path 1464:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
247. public boolean isEmpty() //jndiconfiguration.java
       ...
249. try
```

[Table of Contents] [Description]

Path 1465:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
300. public boolean containsKey(String key) //jndiconfiguration.java
       ...
307. try
```

[Table of Contents] [Description]

Path 1466:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
636. public String getString(String key, String
defaultValue) //jndiconfiguration.java
       ...
638. try
```

[Table of Contents] [Description]

Path 1467:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
702. public Vector getVector(String key, Vector
defaultValue) //jndiconfiguration.java
       ...
704. try
```

[Table of Contents] [Description]

Path 1468:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
742. private Object getValueFromJNDI(String key) //jndiconfiguration.java
       ...
748. try
```

[Table of Contents] [Description]

Path 1469:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
207. public InputStream getPropertyStream(String resourceName) throws
IOException //propertiesconfiguration.java
      ...
      212. try
```

[Table of Contents] [Description]

Path 1470:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
208. Connection getConnection(long time) //dbconnectionmanager.java
      ...
      214. try {
```

[Table of Contents] [Description]

Path 1471:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
272. synchronized void freeConnection(Connection con) //dbconnectionmanager.java
      ...
      281. try {
```

[Table of Contents] [Description]

Path 1472:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
305. synchronized Connection getConnection() //dbconnectionmanager.java
      ...
      313. try {
```

[Table of Contents] [Description]

Path 1473:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
352. Connection getConnection(long timeout) //dbconnectionmanager.java
      ...
      364. try {
```

[Table of Contents] [Description]

Path 1474:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
376. synchronized boolean release() //dbconnectionmanager.java
       ...
381. try {
```

[Table of Contents] [Description]

Path 1475:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
401. private Connection newConnection() //dbconnectionmanager.java
       ...
403. try {
```

[Table of Contents] [Description]

Path 1476:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
52. public final class DBUtils //dbutils.java
       ...
104. try {
```

[Table of Contents] [Description]

Path 1477:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
305. public static void closeConnection(Connection connection) //dbutils.java
       ...
309. try {
```

[Table of Contents] [Description]

Path 1478:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
305. public static void closeConnection(Connection connection) //dbutils.java
       ...
316. try {
```

[Table of Contents] [Description]

Path 1479:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
329. public static void resetStatement(Statement statement) //dbutils.java
     ...
331. try {
```

[Table of Contents] [Description]

Path 1480:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
329. public static void resetStatement(Statement statement) //dbutils.java
     ...
337. try {
```

[Table of Contents] [Description]

Path 1481:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
349. public static void closeStatement(Statement statement) //dbutils.java
     350. try {
```

[Table of Contents] [Description]

Path 1482:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
361. public static void closeResultSet(ResultSet rs) //dbutils.java
     362. try {
```

[Table of Contents] [Description]

Path 1483:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. private static void load() //dbutils2.java
    ...
91. try {
```

[Table of Contents] [Description]

Path 1484:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
229. public static void closeConnection(Connection connection) //dbutils2.java
      |...
      |233. try {
```

[Table of Contents] [Description]

Path 1485:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
229. public static void closeConnection(Connection connection) //dbutils2.java
      |...
      |240. try {
```

[Table of Contents] [Description]

Path 1486:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
45. public static String filter(String input) //htmlnewlinefilter.java
      |...
      |55. try {
```

[Table of Contents] [Description]

Path 1487:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
62. public ResourceBundle getResourceBundle(Locale
locale) //cacheresourcebundle.java
      |...
      |68. try {
```

[Table of Contents] [Description]

Path 1488:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      |73. try {
```

[Table of Contents] [Description]

Path 1489:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
78. try {
```

[Table of Contents] [Description]

Path 1490:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
83. try {
```

[Table of Contents] [Description]

Path 1491:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
88. try {
```

[Table of Contents] [Description]

Path 1492:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
93. try {
```

[Table of Contents] [Description]

Path 1493:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
98. try {
```

[Table of Contents] [Description]

Path 1494:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
103. try {
```

[Table of Contents] [Description]

Path 1495:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
108. try {
```

[Table of Contents] [Description]

Path 1496:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
113. try {
```

[Table of Contents] [Description]

Path 1497:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
118. try {
```

[Table of Contents] [Description]

Path 1498:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
123. try {
```

[Table of Contents] [Description]

Path 1499:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
128. try {
```

[Table of Contents] [Description]

Path 1500:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
133. try {
```

[Table of Contents] [Description]

Path 1501:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
138. try {
```

[Table of Contents] [Description]

Path 1502:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
143. try {
```

[Table of Contents] [Description]

Path 1503:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
148. try {
```

[Table of Contents] [Description]

Path 1504:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
153. try {
```

[Table of Contents] [Description]

Path 1505:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
158. try {
```

[Table of Contents] [Description]

Path 1506:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
163. try {
```

[Table of Contents] [Description]

Path 1507:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
168. try {
```

[Table of Contents] [Description]

Path 1508:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
173. try {
```

[Table of Contents] [Description]

Path 1509:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
178. try {
```

[Table of Contents] [Description]

Path 1510:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
183. try {
```

[Table of Contents] [Description]

Path 1511:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
70. public static synchronized String getMD5_Base64(String input) //encoder.java
      ...
86. try {
```

[Table of Contents] [Description]

Path 1512:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
146. public static String encodeURL(String input) //encoder.java
147. try {
```

[Table of Contents] [Description]

Path 1513:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
160. public static String decodeURL(String input) //encoder.java
161. try {
```

[Table of Contents] [Description]

Path 1514:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public void contextDestroyed(ServletContextEvent
event) //mvncorelifecycleserviceimpldefault.java
```

```
...
87. try {
```

[Table of Contents] [Description]

Path 1515:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
     196. try {
```

[Table of Contents] [Description]

Path 1516:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public int doStartTag() throws JspTagException //basetag.java
     59. try {
```

[Table of Contents] [Description]

Path 1517:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public int doEndTag() throws JspTagException //basetag.java
     74. try {
```

[Table of Contents] [Description]

Path 1518:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
59. public int doStartTag() throws JspTagException //bodytag.java
     60. try {
```

[Table of Contents] [Description]

Path 1519:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
86. public int doEndTag() throws JspTagException //bodytag.java
     87. try {
```

[Table of Contents] [Description]

Path 1520:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
75. public int doStartTag() throws JspTagException //cssrowtag.java
      ...
83. try {
```

[Table of Contents] [Description]

Path 1521:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public int doStartTag() throws JspTagException //framesettag.java
59. try {
```

[Table of Contents] [Description]

Path 1522:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public int doEndTag() throws JspTagException //framesettag.java
74. try {
```

[Table of Contents] [Description]

Path 1523:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public int doStartTag() throws JspTagException //frametag.java
59. try {
```

[Table of Contents] [Description]

Path 1524:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public int doEndTag() throws JspTagException //frametag.java
74. try {
```

[Table of Contents] [Description]

Path 1525:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public int doStartTag() throws JspTagException //headtag.java
      59. try {
```

[Table of Contents] [Description]

Path 1526:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
73. public int doEndTag() throws JspTagException //headtag.java
      74. try {
```

[Table of Contents] [Description]

Path 1527:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
60. public int doStartTag() throws JspTagException //htmltag.java
      61. try {
```

[Table of Contents] [Description]

Path 1528:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
81. public int doEndTag() throws JspTagException //htmltag.java
      82. try {
```

[Table of Contents] [Description]

Path 1529:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
74. public int doStartTag() throws JspTagException //methodtag.java
      ...
      77. try {
```

[Table of Contents] [Description]

Path 1530:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
58. public int doStartTag() throws JspTagException //titletag.java
      59. try {
```

[Table of Contents] [Description]

Path 1531:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
77. public int doEndTag() throws JspTagException //titletag.java
      78. try {
```

[Table of Contents] [Description]

Path 1532:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
53. public JpegEncoder(Image image, int quality, OutputStream
out) //jpegencoder.java
      ...
      57. try {
```

[Table of Contents] [Description]

Path 1533:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
91. public void Compress() //jpegencoder.java
      ...
      95. try {
```

[Table of Contents] [Description]

Path 1534:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
315. void WriteMarker(byte[] data, BufferedOutputStream out) //jpegencoder.java
      316. try {
```

[Table of Contents] [Description]

Path 1535:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
324. void WriteArray(byte[] data, BufferedOutputStream out) //jpegencoder.java
      ...
      326. try {
```

[Table of Contents] [Description]

Path 1536:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
958. void bufferIt(BufferedOutputStream outStream, int code,int
size) //jpegencoder.java
      ...
970. try
```

[Table of Contents] [Description]

Path 1537:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
958. void bufferIt(BufferedOutputStream outStream, int code,int
size) //jpegencoder.java
      ...
979. try
```

[Table of Contents] [Description]

Path 1538:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
996. void flushBuffer(BufferedOutputStream outStream) //jpegencoder.java
      ...
1001. try
```

[Table of Contents] [Description]

Path 1539:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
996. void flushBuffer(BufferedOutputStream outStream) //jpegencoder.java
      ...
1010. try {
```

[Table of Contents] [Description]

Path 1540:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
996. void flushBuffer(BufferedOutputStream outStream) //jpegencoder.java
      ...
1023. try
```

[Table of Contents] [Description]

Path 1541:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
1264. private void getYCCArray() //jpegencoder.java
      ...
1296. try
```

[Table of Contents] [Description]

Path 1542:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
93. public void write(String value) throws IOException //csvutil.java
94. try {
```

[Table of Contents] [Description]

Path 1543:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
136. public void writeln(String value) throws IOException //csvutil.java
137. try {
```

[Table of Contents] [Description]

Path 1544:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
146. public void writeln() throws IOException //csvutil.java
147. try {
```

[Table of Contents] [Description]

Path 1545:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
157. public void writeln(String[] values) throws IOException //csvutil.java
158. try {
```

[Table of Contents] [Description]

Path 1546:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
167. public void write(String[] values) throws IOException //csvutil.java
     168. try {
```

[Table of Contents] [Description]

Path 1547:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
178. public void writeln(String[][] values) throws IOException //csvutil.java
     179. try {
```

[Table of Contents] [Description]

Path 1548:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
192. public void writelnComment(String comment) throws IOException //csvutil.java
     193. try {
```

[Table of Contents] [Description]

Path 1549:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
225. public void println(String value) //csvutil.java
     226. try {
```

[Table of Contents] [Description]

Path 1550:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
233. public void println() //csvutil.java
     234. try {
```

[Table of Contents] [Description]

Path 1551:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
241. public void println(String[] values) //csvutil.java
     242. try {
```

[Table of Contents] [Description]

Path 1552:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
249. public void print(String[] values) //csvutil.java
     250. try {
```

[Table of Contents] [Description]

Path 1553:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
257. public void println(String[][] values) //csvutil.java
     258. try {
```

[Table of Contents] [Description]

Path 1554:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
265. public void printlnComment(String comment) //csvutil.java
     266. try {
```

[Table of Contents] [Description]

Path 1555:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
273. public void print(String value) //csvutil.java
     274. try {
```

[Table of Contents] [Description]

Path 1556:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
285. public boolean checkError() //csvutil.java
     286. try {
```

[Table of Contents] [Description]

Path 1557:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
     ...
244. try {
```

[Table of Contents] [Description]

Path 1558:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
     ...
251. try {
```

[Table of Contents] [Description]

Path 1559:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
298. public static void popFile(File srcFile, OutputStream output) throws
IOException //fileutil.java
     ...
411. public static void writeFile(String content, String fileName, String
destEncoding)
```

[Table of Contents] [Description]

Path 1560:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
381. public static void createTextFile(InputStream inputStream, String
textFile) //fileutil.java
     ...
388. try {
```

[Table of Contents] [Description]

Path 1561:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
     ...
428. try {
```

[Table of Contents] [Description]

Path 1562:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
     ...
441. try {
```

[Table of Contents] [Description]

Path 1563:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
     ...
461. try {
```

[Table of Contents] [Description]

Path 1564:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
     ...
480. try {
```

[Table of Contents] [Description]

Path 1565:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
linesToReturn) //fileutil.java
     ...
522. try {
```

[Table of Contents] [Description]

Path 1566:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
```

```
linesToReturn) //fileutil.java
    ...
531. try {
```

[Table of Contents] [Description]

Path 1567:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
linesToReturn) //fileutil.java
    ...
551. try {
```

[Table of Contents] [Description]

Path 1568:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
577. public static boolean touch(String fileName) //fileutil.java
    578. try {
```

[Table of Contents] [Description]

Path 1569:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
106. try {
```

[Table of Contents] [Description]

Path 1570:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
128. public static int getParameterInt(GenericRequest request, String param, int
defaultValue) //genericparamutil.java
    ...
136. try {
```

[Table of Contents] [Description]

Path 1571:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
158. public static long getParameterLong(GenericRequest request, String
param) //genericparamutil.java
     ...
163. try {
```

[Table of Contents] [Description]

Path 1572:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
173. public static long getParameterLong(GenericRequest request, String param, long
defaultValue) //genericparamutil.java
     ...
182. try {
```

[Table of Contents] [Description]

Path 1573:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
203. public static byte getParameterByte(GenericRequest request, String
param) //genericparamutil.java
     ...
208. try {
```

[Table of Contents] [Description]

Path 1574:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
218. public static double getParameterDouble(GenericRequest request, String
param) //genericparamutil.java
     ...
223. try {
```

[Table of Contents] [Description]

Path 1575:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
284. public static java.sql.Date getParameterDate(GenericRequest request, String
param) //genericparamutil.java
     ...
289. try {
```

[Table of Contents] [Description]

Path 1576:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
302. public static java.util.Date getParameterDateUtil(GenericRequest request,
String param) //genericparamutil.java
     |...
     |307. try {
```

[Table of Contents] [Description]

Path 1577:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
320. public static java.sql.Date getParameterDate(GenericRequest request, String
paramDay, String paramMonth, String paramYear) //genericparamutil.java
     |...
     |331. try {
```

[Table of Contents] [Description]

Path 1578:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public static java.sql.Date getParameterDateSafe(GenericRequest request, String
paramDay, String paramMonth, String paramYear) //genericparamutil.java
     |...
     |347. try {
```

[Table of Contents] [Description]

Path 1579:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
341. public static java.sql.Date getParameterDateSafe(GenericRequest request, String
paramDay, String paramMonth, String paramYear) //genericparamutil.java
     |...
     |359. try {
```

[Table of Contents] [Description]

Path 1580:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
97. public static void createThumbnail(InputStream inputStream, OutputStream
outputStream, int maxWidth, int maxHeight) //imageutil.java
    ...
115. try {
```

[Table of Contents] [Description]

Path 1581:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
97. public static void createThumbnail(InputStream inputStream, OutputStream
outputStream, int maxWidth, int maxHeight) //imageutil.java
    ...
168. try {
```

[Table of Contents] [Description]

Path 1582:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
97. public static void createThumbnail(InputStream inputStream, OutputStream
outputStream, int maxWidth, int maxHeight) //imageutil.java
    ...
175. try {
```

[Table of Contents] [Description]

Path 1583:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
224. public static Image getFitSizeImage(InputStream inputStream, int fitWidth, int
fitHeight) //imageutil.java
    ...
226. try {
```

[Table of Contents] [Description]

Path 1584:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
249. public static void writeJpegImage_Sun(Image image, OutputStream outputStream)
throws IOException //imageutil.java
    ...
279. try {
```

[Table of Contents] [Description]

Path 1585:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
300. public static void writeJpegImage_nonSun(Image image, OutputStream
outputStream) throws IOException //imageutil.java
     ...
310. try {
```

[Table of Contents] [Description]

Path 1586:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
647. public static boolean isClassAccessible(String className) //imageutil.java
     ...
649. try {
```

[Table of Contents] [Description]

Path 1587:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
     ...
194. try {
```

[Table of Contents] [Description]

Path 1588:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
     ...
233. try {
```

[Table of Contents] [Description]

Path 1589:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
     ...
253. try {
```

[Table of Contents] [Description]

Path 1590:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
292. try {
```

[Table of Contents] [Description]

Path 1591:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
353. try {
```

[Table of Contents] [Description]

Path 1592:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
390. try {
```

[Table of Contents] [Description]

Path 1593:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
473. public static MailMessageStruct[] receiveEmail(String popServer, String
popUser, String popPassword) //mailutil.java
       ...
518. try {
```

[Table of Contents] [Description]

Path 1594:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
473. public static MailMessageStruct[] receiveEmail(String popServer, String
popUser, String popPassword) //mailutil.java
       ...
525. try {
```

[Table of Contents] [Description]

Path 1595:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
144. public static int getParameterInt(HttpServletRequest request, String
param) //paramutil.java
     ...
149. try {
```

[Table of Contents] [Description]

Path 1596:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
171. public static int getParameterInt(HttpServletRequest request, String param, int
defaultValue) //paramutil.java
     ...
179. try {
```

[Table of Contents] [Description]

Path 1597:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
201. public static long getParameterLong(HttpServletRequest request, String
param) //paramutil.java
     ...
206. try {
```

[Table of Contents] [Description]

Path 1598:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
216. public static long getParameterLong(HttpServletRequest request, String param,
long defaultValue) //paramutil.java
     ...
225. try {
```

[Table of Contents] [Description]

Path 1599:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
246. public static byte getParameterByte(HttpServletRequest request, String
param) //paramutil.java
      ...
251. try {
```

[Table of Contents] [Description]

Path 1600:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
261. public static double getParameterDouble(HttpServletRequest request, String
param) //paramutil.java
      ...
266. try {
```

[Table of Contents] [Description]

Path 1601:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
325. public static java.sql.Date getParameterDate(HttpServletRequest request, String
param) //paramutil.java
      ...
330. try {
```

[Table of Contents] [Description]

Path 1602:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
343. public static java.util.Date getParameterDateUtil(HttpServletRequest request,
String param) //paramutil.java
      ...
348. try {
```

[Table of Contents] [Description]

Path 1603:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
361. public static java.sql.Date getParameterDate(HttpServletRequest request, String
paramDay, String paramMonth, String paramYear) //paramutil.java
      ...
372. try {
```

[Table of Contents] [Description]

Path 1604:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
94. public void schedule(TimerTask task, Date firstTime, long
period) //timerutil.java
      ...
96. try {
```

[Table of Contents] [Description]

Path 1605:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
105. public void schedule(TimerTask task, Date time) //timerutil.java
      ...
107. try {
```

[Table of Contents] [Description]

Path 1606:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
116. public void schedule(TimerTask task, long delay) //timerutil.java
      ...
118. try {
```

[Table of Contents] [Description]

Path 1607:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
127. public void schedule(TimerTask task, long delay, long period) //timerutil.java
      ...
129. try {
```

[Table of Contents] [Description]

Path 1608:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
138. public void scheduleAtFixedRate(TimerTask task, Date firstTime, long
period) //timerutil.java
      ...
140. try {
```

[Table of Contents] [Description]

Path 1609:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
149. public void scheduleAtFixedRate(TimerTask task, long delay, long
period) //timerutil.java
      |
      ...
      151. try {
```

[Table of Contents] [Description]

Path 1610:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
283. public List /* FileItem */ parseRequest(RequestContext
ctx) //fileuploadbase.java
      |
      ...
      321. try {
```

[Table of Contents] [Description]

Path 1611:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
404. protected byte[] getBoundary(String contentType) //fileuploadbase.java
      |
      ...
      415. try {
```

[Table of Contents] [Description]

Path 1612:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
519. protected Map /* String, String */ parseHeaders(String
headerPart) //fileuploadbase.java
      |
      ...
      526. try {
```

[Table of Contents] [Description]

Path 1613:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
357. public boolean readBoundary() //multipartstream.java
      |
```

```
...
363. try {
```

[Table of Contents] [Description]

Path 1614:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
436. public String readHeaders() //multipartstream.java
...
445. try {
```

[Table of Contents] [Description]

Path 1615:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
436. public String readHeaders() //multipartstream.java
...
463. try {
```

[Table of Contents] [Description]

Path 1616:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
620. public boolean skipPreamble() //multipartstream.java
...
625. try {
```

[Table of Contents] [Description]

Path 1617:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
295. public byte[] get() //diskfileitem.java
...
306. try {
```

[Table of Contents] [Description]

Path 1618:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
295. public byte[] get() //diskfileitem.java
```

```
...
313. try {
```

[Table of Contents] [Description]

Path 1619:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
352. public String getString() //diskfileitem.java
...
358. try {
```

[Table of Contents] [Description]

Path 1620:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
386. public void write(File file) throws Exception //diskfileitem.java
...
418. try {
```

[Table of Contents] [Description]

Path 1621:

Query Name - **Catch_Without_General_Exception_Handling**

Severity - **Information**

```
386. public void write(File file) throws Exception //diskfileitem.java
...
425. try {
```

[Table of Contents] [Description]

Incorrect Block Delimitation

CWE ID	483
Description	In some languages, forgetting to explicitly delimit a block can result in a logic error that can, in turn, have security implications.
Likelihood of Exploit	Low
Common Consequences	This is a general logic error -- with all the potential consequences that this entails.
Potential Mitigations	Implementation: Always use explicit block delimitation and use static-analysis technologies to enforce this practice.
Demonstrative Examples	In this example, when the condition is true, the intention may be that both x and y run. if (condition==true) x; y;
Context Notes	In many languages, braces are optional for blocks, and --

	in a case where braces are omitted -- it is possible to insert a logic error where a statement is thought to be in a block but is not. This is a common and well known reliability error.
Node Relationships	ChildOf - Code Quality (398)
Source Taxonomies	CLASP - Incorrect block delimitation
Applicable Platforms	All

Path 1:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
46. public String getMD5_Base64(String input) //mvntomcatjdbcrealm.java
      ...
56. if (digest == null)
```

[Table of Contents] [Description]

Path 2:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
150. public static int correctMvnforumMemberID(int phpbbUserID) //migrator.java
151. if (phpbbUserID == PHPBB_ADMIN_ID)
```

[Table of Contents] [Description]

Path 3:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
150. public static int correctMvnforumMemberID(int phpbbUserID) //migrator.java
      ...
153. else if (phpbbUserID == PHPBB_GUEST_ID)
```

[Table of Contents] [Description]

Path 4:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
430. public static void migratePosts(PrintStream ostream) //migrator.java
      ...
458. if (ppidtmp == postID)
```

[Table of Contents] [Description]

Path 5:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
125. public static synchronized String getMD5_Base64(String input) //md5.java
       ...
137. if (digest == null)
```

[Table of Contents] [Description]

Path 6:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
120. public void exportGroupList() //importexportxml.java
       ...
137. if (groupOwnerID == 0)
```

[Table of Contents] [Description]

Path 7:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
       ...
262. if (ppidtmp == postID)
```

[Table of Contents] [Description]

Path 8:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
       ...
376. if (phpbbGroup == null) continue;
```

[Table of Contents] [Description]

Path 9:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
179. protected void handleParseConversionException(Exception e) throws
SAXException //abstractunmarshallingeventhandlerimpl.java
       180. if( e instanceof RuntimeException )
```

[Table of Contents] [Description]

Path 10:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
80. public synchronized com.sun.msv.grammar.Grammar getGrammar() throws
JAXBException //defaultjaxbcontextimpl.java
      |81. if( grammar==null )
```

[Table of Contents] [Description]

Path 11:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
132. public Object newInstance( Class
javaContentInterface ) //defaultjaxbcontextimpl.java
      |...
      |141. if(c==null)
```

[Table of Contents] [Description]

Path 12:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public void leaveElement(String uri, String local, String qname) throws
SAXException //discarder.java
      |...
      |43. if(depth==0)
```

[Table of Contents] [Description]

Path 13:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public void leaveElement(String uri, String local, String qname) throws
SAXException //discarder.java
      |...
      |43. if(depth==0)
```

[Table of Contents] [Description]

Path 14:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
127. static GrammarInfo createGrammarInfoFacade( String
contextPath, //grammarinfofacade.java
      |...
      |148. if( version==null ) version = getVersion(c);
```

[Table of Contents] [Description]

Path 15:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
127. static GrammarInfo createGrammarInfoFacade( String
contextPath, //grammarinfofacade.java
     ...
181. if( gis.size()==1 )
```

[Table of Contents] [Description]

Path 16:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
208. public Class getDefaultImplementation( Class
javaContentInterface ) //grammarinfofacade.java
     ...
211. if(c!=null) return c;
```

[Table of Contents] [Description]

Path 17:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
218. public com.sun.msv.grammar.Grammar getGrammar() throws
JAXBException //grammarinfofacade.java
     ...
234. if( bgm == null ) bgm = n;
```

[Table of Contents] [Description]

Path 18:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
68. private final Class lookupRootMap( String nsUri, String
localName ) //grammarinfoimpl.java
     ...
73. if(rootTagMap.containsKey(qn)) return (Class)rootTagMap.get(qn);
```

[Table of Contents] [Description]

Path 19:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
68. private final Class lookupRootMap( String nsUri, String
localName ) //grammarinfoimpl.java
     ...
76. if(rootTagMap.containsKey(qn)) return (Class)rootTagMap.get(qn);
```

[Table of Contents] [Description]

Path 20:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
82. public final Class getRootElement(String namespaceUri, String
localName) //grammarinfoimpl.java
      ...
84. if(intfCls==null) return null;
```

[Table of Contents] [Description]

Path 21:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public final UnmarshallingEventHandler createUnmarshaller
( //grammarinfoimpl.java
      ...
92. if(impl==null) return null;
```

[Table of Contents] [Description]

Path 22:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
40. public boolean contains(Object key) //identityhashset.java
      ...
46. if (e == null)
```

[Table of Contents] [Description]

Path 23:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
40. public boolean contains(Object key) //identityhashset.java
      ...
48. if (e==key)
```

[Table of Contents] [Description]

Path 24:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
84. public boolean add(Object newObj) //identityhashset.java
      85. if (count >= threshold)
```

[Table of Contents] [Description]

Path 25:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
84. public boolean add(Object newObj) //identityhashset.java
      85. if (count >= threshold)
```

[Table of Contents] [Description]

Path 26:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
84. public boolean add(Object newObj) //identityhashset.java
      ...
      94. if(existing==newObj) return false;
```

[Table of Contents] [Description]

Path 27:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
      87. if(so==null)
```

[Table of Contents] [Description]

Path 28:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
      120. if (sr.getWriter() != null)
```

[Table of Contents] [Description]

Path 29:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
      122. else if (sr.getOutputStream() != null)
```

[Table of Contents] [Description]

Path 30:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
128. if (fileURL.substring(8).indexOf(":") > 0)
```

[Table of Contents] [Description]

Path 31:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
141. if (w == null)
```

[Table of Contents] [Description]

Path 32:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
187. protected CharacterEscapeHandler createEscapeHandler( String
encoding ) //marshallerimpl.java
      188. if( escapeHandler!=null )
```

[Table of Contents] [Description]

Path 33:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
187. protected CharacterEscapeHandler createEscapeHandler( String
encoding ) //marshallerimpl.java
      ...
192. if( encoding.startsWith("UTF") )
```

[Table of Contents] [Description]

Path 34:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public Object getProperty(String name) throws
PropertyException //marshallerimpl.java
```

```
250. if( INDENT_STRING.equals(name) )
```

[Table of Contents] [Description]

Path 35:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public Object getProperty(String name) throws
PropertyException //marshallerimpl.java
     ...
252. if( ENCODING_HANDLER.equals(name) )
```

[Table of Contents] [Description]

Path 36:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public Object getProperty(String name) throws
PropertyException //marshallerimpl.java
     ...
254. if( PREFIX_MAPPER.equals(name) )
```

[Table of Contents] [Description]

Path 37:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public Object getProperty(String name) throws
PropertyException //marshallerimpl.java
     ...
256. if( XMLDECLARATION.equals(name) )
```

[Table of Contents] [Description]

Path 38:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public Object getProperty(String name) throws
PropertyException //marshallerimpl.java
     ...
258. if( XML_HEADERS.equals(name) )
```

[Table of Contents] [Description]

Path 39:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
138. public final void text( String text, String fieldName ) throws
SAXException //msvvalidator.java
      |
      ...
      144. if(buf.length()!=0)
```

[Table of Contents] [Description]

Path 40:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
138. public final void text( String text, String fieldName ) throws
SAXException //msvvalidator.java
      |
      ...
      144. if(buf.length()!=0)
```

[Table of Contents] [Description]

Path 41:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
186. private void writePendingText() throws SAXException //msvvalidator.java
      |
      ...
      196. if(buf.length()>1024)
```

[Table of Contents] [Description]

Path 42:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
186. private void writePendingText() throws SAXException //msvvalidator.java
      |
      ...
      196. if(buf.length()>1024)
```

[Table of Contents] [Description]

Path 43:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
282. public void childAsBody( JAXBObject o, String fieldName ) throws
SAXException //msvvalidator.java
      |
      ...
      291. if( insideAttribute ) childAsAttributeBody(vo,fieldName);
```

[Table of Contents] [Description]

Path 44:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
282. public void childAsBody( JAXBObject o, String fieldName ) throws
SAXException //msvvalidator.java
      |
      ...
      291. if( insideAttribute ) childAsAttributeBody(vo,fieldName);
```

[Table of Contents] [Description]

Path 45:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
282. public void childAsBody( JAXBObject o, String fieldName ) throws
SAXException //msvvalidator.java
      |
      ...
      291. if( insideAttribute ) childAsAttributeBody(vo,fieldName);
```

[Table of Contents] [Description]

Path 46:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
295. private void childAsElementBody( Object o, ValidatableObject vo ) throws
SAXException //msvvalidator.java
      |
      ...
      341. if(!child.onAttribute2(
```

[Table of Contents] [Description]

Path 47:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
295. private void childAsElementBody( Object o, ValidatableObject vo ) throws
SAXException //msvvalidator.java
      |
      ...
      341. if(!child.onAttribute2(
```

[Table of Contents] [Description]

Path 48:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
      ...
      90. if( !requirePrefix && nss.getURI("").equals(namespaceUri) )
```

[Table of Contents] [Description]

Path 49:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
      |...
      |97. if (requirePrefix)
```

[Table of Contents] [Description]

Path 50:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
      |...
      |102. if( requirePrefix && namespaceUri.length()==0 )
```

[Table of Contents] [Description]

Path 51:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
      |...
      |129. if( prefix==null )
```

[Table of Contents] [Description]

Path 52:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
      |...
      |138. if( prefixMapper!=null )
```

[Table of Contents] [Description]

Path 53:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
      |
```

```
...
144. if( prefix==null )
```

[Table of Contents] [Description]

Path 54:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public String declareNamespace( String namespaceUri, String preferedPrefix,
boolean requirePrefix ) //namespacecontextimpl.java
...
152. if( requirePrefix && prefix.length()==0 )
```

[Table of Contents] [Description]

Path 55:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public String getNamespaceURI( String prefix ) //namespacecontextimpl.java
...
211. if(uri!=null) return uri;
```

[Table of Contents] [Description]

Path 56:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
216. public Iterator getPrefixes( String namespaceUri ) //namespacecontextimpl.java
...
221. if(prefix!=null) s.add(prefix);
```

[Table of Contents] [Description]

Path 57:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
216. public Iterator getPrefixes( String namespaceUri ) //namespacecontextimpl.java
...
221. if(prefix!=null) s.add(prefix);
```

[Table of Contents] [Description]

Path 58:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
216. public Iterator getPrefixes( String namespaceUri ) //namespacecontextimpl.java
     ...
223. if( nss.getURI("").equals(namespaceUri) )
```

[Table of Contents] [Description]

Path 59:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
216. public Iterator getPrefixes( String namespaceUri ) //namespacecontextimpl.java
     ...
223. if( nss.getURI("").equals(namespaceUri) )
```

[Table of Contents] [Description]

Path 60:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
251. public void endNamespaceDecls() //namespacecontextimpl.java
     ...
258. if(!uri.equals(nss.getURI(prefix))) // avoid redundant decls.
```

[Table of Contents] [Description]

Path 61:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
251. public void endNamespaceDecls() //namespacecontextimpl.java
     ...
258. if(!uri.equals(nss.getURI(prefix))) // avoid redundant decls.
```

[Table of Contents] [Description]

Path 62:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
117. public void startElement( String uri, String local ) throws
SAXException //saxmarshaller.java
     ...
139. if( uris[i]!=null )
```

[Table of Contents] [Description]

Path 63:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
117. public void startElement( String uri, String local ) throws
SAXException //saxmarshaller.java
      |
      ...
      139. if( uris[i]!=null )
```

[Table of Contents] [Description]

Path 64:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
166. public void endAttributes() throws SAXException //saxmarshaller.java
      |
      ...
      175. if(prefix.length()!=0 )
```

[Table of Contents] [Description]

Path 65:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
198. public void endElement() throws SAXException //saxmarshaller.java
      |
      ...
      208. if(prefix.length()!=0)
```

[Table of Contents] [Description]

Path 66:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
268. public void text( String text, String fieldName ) throws
SAXException //saxmarshaller.java
      |
      ...
      276. if(textBuf.length()!=0)
```

[Table of Contents] [Description]

Path 67:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
268. public void text( String text, String fieldName ) throws
SAXException //saxmarshaller.java
      |
      ...
      276. if(textBuf.length()!=0)
```

[Table of Contents] [Description]

Path 68:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
285. private void writePendingText() throws SAXException //saxmarshaller.java
      ...
289. if(len!=0)
```

[Table of Contents] [Description]

Path 69:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
285. private void writePendingText() throws SAXException //saxmarshaller.java
      ...
289. if(len!=0)
```

[Table of Contents] [Description]

Path 70:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
425. public void reportError( ValidationEvent ve ) throws
AbortSerializationException //saxmarshaller.java
      ...
435. if(ve.getLinkedException() instanceof Exception)
```

[Table of Contents] [Description]

Path 71:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
448. private static void _assert( boolean b ) //saxmarshaller.java
449. if(!b)
```

[Table of Contents] [Description]

Path 72:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
458. public String getPreferredPrefix(String namespaceUri, String suggestion,
boolean requirePrefix) //saxmarshaller.java
459. if( namespaceUri.equals("http://www.w3.org/2001/XMLSchema-instance") )
```

[Table of Contents] [Description]

Path 73:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
91. public void startElement( String uri, String local, String qname, Attributes
atts ) //saxunmarshallerhandlerimpl.java
      ...
95. if( uri==null )
```

[Table of Contents] [Description]

Path 74:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
91. public void startElement( String uri, String local, String qname, Attributes
atts ) //saxunmarshallerhandlerimpl.java
      ...
97. if( local==null || local.length()==0 )
```

[Table of Contents] [Description]

Path 75:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
91. public void startElement( String uri, String local, String qname, Attributes
atts ) //saxunmarshallerhandlerimpl.java
      ...
99. if( qname==null || qname.length()==0 )
```

[Table of Contents] [Description]

Path 76:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public final void endElement( String uri, String local, String
qname ) //saxunmarshallerhandlerimpl.java
      ...
133. if( uri==null )
```

[Table of Contents] [Description]

Path 77:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public final void endElement( String uri, String local, String
qname ) //saxunmarshallerhandlerimpl.java
      ...
135. if( local==null || local.length()==0 )
```

[Table of Contents] [Description]

Path 78:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public final void endElement( String uri, String local, String
qname ) //saxunmarshallerhandlerimpl.java
      |...
      |137. if( qname==null || qname.length()==0 )
```

[Table of Contents] [Description]

Path 79:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
150. public Object getResult() throws
UnmarshalException //saxunmarshallerhandlerimpl.java
      |151. if(isUnmarshalInProgress)
```

[Table of Contents] [Description]

Path 80:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
150. public Object getResult() throws
UnmarshalException //saxunmarshallerhandlerimpl.java
      |...
      |154. if(!aborted) return result;
```

[Table of Contents] [Description]

Path 81:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
204. protected void consumeText( String str, boolean ignorable ) throws
SAXException //saxunmarshallerhandlerimpl.java
      |205. if(ignorable && str.trim().length()==0)
```

[Table of Contents] [Description]

Path 82:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. private void processText( boolean ignorable ) throws
SAXException //saxunmarshallerhandlerimpl.java
      |
```

```
214. if( shouldCollectText() )
```

[Table of Contents] [Description]

Path 83:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. private void processText( boolean ignorable ) throws
SAXException //saxunmarshallerhandlerimpl.java
214. if( shouldCollectText() )
```

[Table of Contents] [Description]

Path 84:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. private void processText( boolean ignorable ) throws
SAXException //saxunmarshallerhandlerimpl.java
...
219. if(buffer.length()<1024) buffer.setLength(0);
```

[Table of Contents] [Description]

Path 85:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
213. private void processText( boolean ignorable ) throws
SAXException //saxunmarshallerhandlerimpl.java
...
219. if(buffer.length()<1024) buffer.setLength(0);
```

[Table of Contents] [Description]

Path 86:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
223. public final void characters( char[] buf, int start, int
len ) //saxunmarshallerhandlerimpl.java
224. if( shouldCollectText() )
```

[Table of Contents] [Description]

Path 87:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
223. public final void characters( char[] buf, int start, int
```

```
len ) //saxunmarshallerhandlerimpl.java
      224. if( shouldCollectText() )
```

[Table of Contents] [Description]

Path 88:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public String resolveNamespacePrefix( String
prefix ) //saxunmarshallerhandlerimpl.java
      262. if(prefix.equals("xml"))
```

[Table of Contents] [Description]

Path 89:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public String resolveNamespacePrefix( String
prefix ) //saxunmarshallerhandlerimpl.java
      ...
      266. if(prefix.equals(nsBind[i]))
```

[Table of Contents] [Description]

Path 90:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
298. private List getAllPrefixesInList(String uri) //saxunmarshallerhandlerimpl.java
      ...
      309. if( uri==null )
```

[Table of Contents] [Description]

Path 91:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
321. public String getPrefix(String uri) //saxunmarshallerhandlerimpl.java
     322. if( uri.equals(XMLConstants.XML_NS_URI) )
```

[Table of Contents] [Description]

Path 92:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
321. public String getPrefix(String uri) //saxunmarshallerhandlerimpl.java
     ...
     324. if( uri.equals(XMLConstants.XMLNS_ATTRIBUTE_NS_URI) )
```

[Table of Contents] [Description]

Path 93:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
321. public String getPrefix(String uri) //saxunmarshallerhandlerimpl.java
     ...
     326. if( uri==null )
```

[Table of Contents] [Description]

Path 94:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
338. public String getNamespaceURI(String prefix) //saxunmarshallerhandlerimpl.java
     339. if( prefix==null )
```

[Table of Contents] [Description]

Path 95:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
338. public String getNamespaceURI(String prefix) //saxunmarshallerhandlerimpl.java
     ...
     341. if( prefix.equals(XMLConstants.XMLNS_ATTRIBUTE) )
```

[Table of Contents] [Description]

Path 96:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      |
      ...
      395. if( a==null )
```

[Table of Contents] [Description]

Path 97:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      |
      ...
      395. if( a==null )
```

[Table of Contents] [Description]

Path 98:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      |
      ...
      410. if( auri==null )
```

[Table of Contents] [Description]

Path 99:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      |
      ...
      412. if( alocal==null || alocal.length()==0 )
```

[Table of Contents] [Description]

Path 100:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      |
      ...
      414. if( aqname==null || aqname.length()==0 )
```

[Table of Contents] [Description]

Path 101:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
      ...
425. if(v.equals("false") || v.equals("0"))
```

[Table of Contents] [Description]

Path 102:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
500. public void addPatcher( Runnable job ) //saxunmarshallerhandlerimpl.java
      ...
502. if( patchers==null )
```

[Table of Contents] [Description]

Path 103:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
523. public String addToIdTable( String id ) //saxunmarshallerhandlerimpl.java
      524. if(idmap==null) idmap = new Hashtable();
```

[Table of Contents] [Description]

Path 104:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
529. public Object getObjectFromId( String id ) //saxunmarshallerhandlerimpl.java
      530. if(idmap==null) return null;
```

[Table of Contents] [Description]

Path 105:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
564. public void handleEvent(ValidationEvent event, boolean canRecover ) throws
SAXException //saxunmarshallerhandlerimpl.java
      ...
577. if(!recover) aborted = true;
```

[Table of Contents] [Description]

Path 106:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
564. public void handleEvent(ValidationEvent event, boolean canRecover ) throws
SAXException //saxunmarshallerhandlerimpl.java
      ...
579. if( !canRecover || !recover )
```

[Table of Contents] [Description]

Path 107:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
604. public Tracer getTracer() //saxunmarshallerhandlerimpl.java
      605. if(tracer==null)
```

[Table of Contents] [Description]

Path 108:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
613. private String computeExpectedRootElements() //saxunmarshallerhandlerimpl.java
      ...
619. if(r.length()!=0) r+=',';
```

[Table of Contents] [Description]

Path 109:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
58. public void setValidating(boolean validating) throws
JAXBException //unmarshallerimpl.java
      ...
60. if(validating==true)
```

[Table of Contents] [Description]

Path 110:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
58. public void setValidating(boolean validating) throws
JAXBException //unmarshallerimpl.java
      ...
60. if(validating==true)
```

[Table of Contents] [Description]

Path 111:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
158. public final Object unmarshal( Node node ) throws
JAXBException //unmarshallerimpl.java
      |
      ...
      164. if(node instanceof Element)
```

[Table of Contents] [Description]

Path 112:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
158. public final Object unmarshal( Node node ) throws
JAXBException //unmarshallerimpl.java
      |
      ...
      164. if(node instanceof Element)
```

[Table of Contents] [Description]

Path 113:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
158. public final Object unmarshal( Node node ) throws
JAXBException //unmarshallerimpl.java
      |
      ...
      167. if(node instanceof Document)
```

[Table of Contents] [Description]

Path 114:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
158. public final Object unmarshal( Node node ) throws
JAXBException //unmarshallerimpl.java
      |
      ...
      167. if(node instanceof Document)
```

[Table of Contents] [Description]

Path 115:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
30. public static void handlePrintConversionException( //util.java
      |
```

```
...
33. if( e instanceof SAXException )
```

[Table of Contents] [Description]

Path 116:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
67. private static String getUserFriendlyTypeName( Object o ) //util.java
    68. if( o instanceof ValidatableObject )
```

[Table of Contents] [Description]

Path 117:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
83. public void startElement( String nsUri, String local, String qname, Attributes
atts ) throws SAXException //validatingunmarshaller.java
    ...
    90. if(aUri.equals("http://www.w3.org/2001/XMLSchema-instance")
```

[Table of Contents] [Description]

Path 118:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public String onID( XMLSerializable owner, String value ) throws
SAXException //validationcontext.java
    ...
    108. if(!validateID) return value;
```

[Table of Contents] [Description]

Path 119:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
120. public String onIDREF( XMLSerializable referer, String value ) throws
SAXException //validationcontext.java
    121. if(!validateID) return value;
```

[Table of Contents] [Description]

Path 120:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
120. public String onIDREF( XMLSerializable referer, String value ) throws
SAXException //validationcontext.java
      |...
      |123. if(IDs.contains(value))
```

[Table of Contents] [Description]

Path 121:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
132. protected void reconcileIDs() throws SAXException //validationcontext.java
      |133. if(!validateID) return;
```

[Table of Contents] [Description]

Path 122:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
132. protected void reconcileIDs() throws SAXException //validationcontext.java
      |...
      |138. if(IDs.contains(e.getKey()))
```

[Table of Contents] [Description]

Path 123:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
107. private boolean validate( Object o, boolean validateId ) //validatorimpl.java
      |...
      |115. if(vo==null)
```

[Table of Contents] [Description]

Path 124:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
59. public static String wrapit(String text) //importexportutil.java
      |...
      |63. if (text.charAt(i) == '\'')
```

[Table of Contents] [Description]

Path 125:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
59. public static String wrapit(String text) //importexportutil.java
      ...
      63. if (text.charAt(i) == '\'')
```

[Table of Contents] [Description]

Path 126:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
59. public static String wrapit(String text) //importexportutil.java
      ...
      63. if (text.charAt(i) == '\'')
```

[Table of Contents] [Description]

Path 127:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
119. public static String HexIPtoString(String hexrep) //importexportutil.java
      120. if (hexrep.length() < 8)
```

[Table of Contents] [Description]

Path 128:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
151. public static String stripPHPBBQuotes(String in) //importexportutil.java
      ...
      188. if (err)
```

[Table of Contents] [Description]

Path 129:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
159. public static synchronized String getMD5_Base64(String input) //md5.java
      ...
      171. if (digest == null) return input;
```

[Table of Contents] [Description]

Path 130:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
184. static int intfromchar(char c) //md5.java
```

```
...
189. if(clower == carr[i])
```

[Table of Contents] [Description]

Path 131:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
501. if(ppidtmp.equals(PostID))
```

[Table of Contents] [Description]

Path 132:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
135. public static String wrapit(String text) //utils.java
...
141. if(text.charAt(i) == '\'' )
```

[Table of Contents] [Description]

Path 133:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
135. public static String wrapit(String text) //utils.java
...
141. if(text.charAt(i) == '\'' )
```

[Table of Contents] [Description]

Path 134:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
135. public static String wrapit(String text) //utils.java
...
141. if(text.charAt(i) == '\'' )
```

[Table of Contents] [Description]

Path 135:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
193. public static String HexIPtoString(String hexrep) //utils.java
     |...
     |195. if(hexrep.length() < 8)
```

[Table of Contents] [Description]

Path 136:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
227. public static String stripPHPBBQuotes(String in) //utils.java
     |...
     |271. if(err)
```

[Table of Contents] [Description]

Path 137:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
181. static String convertLine(String line, String startToken, String endToken)
throws IllegalArgumentException //converturl.java
     |...
     |193. if (endIndex < 0) break;
```

[Table of Contents] [Description]

Path 138:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
311. public static void main(String[] args) throws Exception //converturl.java
     |...
     |345. if (dir.isFile()) throw new IOException("IOException ->
     |BadInputException: not a directory.");
```

[Table of Contents] [Description]

Path 139:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      |1827. if (thumbnailWidth < 0) thumbnailWidth = 0;
```

[Table of Contents] [Description]

Path 140:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      1829. if (thumbnailHeight < 0) thumbnailHeight = 0;
```

[Table of Contents] [Description]

Path 141:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      1887. if (maxAttachmentSize < -1) maxAttachmentSize = 0; // -1 is a valid
      value in common-upload, mean no maximum file size
```

[Table of Contents] [Description]

Path 142:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      1890. if (maxMessageAttachmentSize < -1) maxMessageAttachmentSize = 0; // -1
      is a valid value in common-upload, mean no maximum file size
```

[Table of Contents] [Description]

Path 143:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      1893. if (maxFavoriteThreads < 0) maxFavoriteThreads = 0;
```

[Table of Contents] [Description]

Path 144:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      |...
      1896. if (maxPrivateMessages < 0) maxPrivateMessages = 0;
```

[Table of Contents] [Description]

Path 145:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1899. if (maxEditDays < 0) maxEditDays = 0;
```

[Table of Contents] [Description]

Path 146:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1902. if (maxAttachDays < 0) maxAttachDays = 0;
```

[Table of Contents] [Description]

Path 147:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1905. if (maxDeleteDays < 0) maxDeleteDays = 0;
```

[Table of Contents] [Description]

Path 148:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1908. if (rowsPerPage < 5) rowsPerPage = 5;
```

[Table of Contents] [Description]

Path 149:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1911. if (rowsPerRSS < 5) rowsPerRSS = 5;
```

[Table of Contents] [Description]

Path 150:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1914. if (maxHotTopics < 5) maxHotTopics = 5;
```

[Table of Contents] [Description]

Path 151:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1917. if (maxHttpRequestsPerHourPerIP < 0) maxHttpRequestsPerHourPerIP = 0;
```

[Table of Contents] [Description]

Path 152:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1920. if (maxPostsPerHourPerIP < 0) maxPostsPerHourPerIP = 0;
```

[Table of Contents] [Description]

Path 153:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1923. if (maxPostsPerHourPerMember < 0) maxPostsPerHourPerMember = 0;
```

[Table of Contents] [Description]

Path 154:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1926. if (maxMembersPerHourPerIP < 0) maxMembersPerHourPerIP = 0;
```

[Table of Contents] [Description]

Path 155:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1929. if (maxLoginsPerHourPerIP < 0) maxLoginsPerHourPerIP = 0;
```

[Table of Contents] [Description]

Path 156:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1932. if (maxMessagesPerHourPerIP < 0) maxMessagesPerHourPerIP = 0;
```

[Table of Contents] [Description]

Path 157:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1935. if (maxPasswordDays < 0) maxPasswordDays = 1;
```

[Table of Contents] [Description]

Path 158:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1938. if (maxCharsInShortSummary <= 0) maxCharsInShortSummary =
      Integer.MAX_VALUE;
```

[Table of Contents] [Description]

Path 159:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1941. if (maxCharsInLongSummary <= 0) maxCharsInLongSummary =
      Integer.MAX_VALUE;
```

[Table of Contents] [Description]

Path 160:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1944. if (maxCharsInRSS <= 0) maxCharsInRSS = Integer.MAX_VALUE;
```

[Table of Contents] [Description]

Path 161:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1950. if (defaultWatchOption < WatchBean.WATCH_OPTION_DEFAULT ||
      defaultWatchOption > WatchBean.WATCH_OPTION_WEEKLY) defaultWatchOption =
      WatchBean.WATCH_OPTION_DEFAULT;
```

[Table of Contents] [Description]

Path 162:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1953. if (defaultModerationOption <
      ForumBean.FORUM_MODERATION_MODE_SYSTEM_DEFAULT || defaultModerationOption >
      ForumBean.FORUM_MODERATION_MODE_POST_ONLY) defaultModerationOption =
      ForumBean.FORUM_MODERATION_MODE_SYSTEM_DEFAULT;
```

[Table of Contents] [Description]

Path 163:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1956. if (defaultWatchType < WatchBean.WATCH_TYPE_DIGEST || defaultWatchType >
      WatchBean.WATCH_TYPE_NONDIGEST) defaultWatchType =
      WatchBean.WATCH_TYPE_DIGEST;
```

[Table of Contents] [Description]

Path 164:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
470. public static void checkClassName(Locale locale, String className, boolean
```

```
required) throws BadInputException //myutil.java
      ...
      472. if (className.length() == 0) return;
```

[Table of Contents] [Description]

Path 165:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
109. public void setParentPostID(int value) //attachmentxml.java
      110. if (value<0) parentPostID=-1;
```

[Table of Contents] [Description]

Path 166:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
123. public void setParentThreadID(int value) //attachmentxml.java
      124. if (value<0) parentThreadID=-1;
```

[Table of Contents] [Description]

Path 167:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
137. public void setParentForumID(int value) //attachmentxml.java
      138. if (value<0) parentForumID=-1;
```

[Table of Contents] [Description]

Path 168:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
151. public void setParentCategoryID(int value) //attachmentxml.java
      152. if (value<0) parentCategoryID=-1;
```

[Table of Contents] [Description]

Path 169:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
226. public void addAttachment(String strAttachmentID, //attachmentxml.java
      ...
      265. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 170:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
226. public void addAttachment(String strAttachmentID, //attachmentxml.java
     ...
267. if (attachDesc==null) attachDesc="";
```

[Table of Contents] [Description]

Path 171:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
226. public void addAttachment(String strAttachmentID, //attachmentxml.java
     ...
268. if (attachCreationIP==null) attachCreationIP="0.0.0.0";
```

[Table of Contents] [Description]

Path 172:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public void setParentCategoryID(int value) //categoryxml.java
    89. if (value < 0) parentCategoryID = -1;
```

[Table of Contents] [Description]

Path 173:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
113. public void addCategory(String categoryName, //categoryxml.java
     ...
131. if (categoryDesc == null) categoryDesc = "";
```

[Table of Contents] [Description]

Path 174:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
194. public void addCategoryWatch(String memberName, //categoryxml.java
     ...
213. if (memberName == null) memberName = "";
```

[Table of Contents] [Description]

Path 175:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
162. if (len > 0)
```

[Table of Contents] [Description]

Path 176:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
      ...
162. if (len > 0)
```

[Table of Contents] [Description]

Path 177:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
220. private static void addDirectory(ZipOutputStream out, File dir, String
relativePath) //exportwebhelper.java
      ...
227. if (!relativePath.endsWith("/")) relativePath+="/";
```

[Table of Contents] [Description]

Path 178:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
91. public void setParentThreadID(int value) //favoritethreadxml.java
      92. if (value<0) parentThreadID=-1;
```

[Table of Contents] [Description]

Path 179:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
105. public void setParentForumID(int value) //favoritethreadxml.java
      106. if (value<0) parentForumID=-1;
```

[Table of Contents] [Description]

Path 180:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
88. public void setParentCategoryID(int value) //forumxml.java
      89. if (value<0) parentCategoryID=-1;
```

[Table of Contents] [Description]

Path 181:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
121. public void addForum(String lastPostMemberName, String
forumName, //forumxml.java
      ...
      150. if (lastPostMemberName==null) lastPostMemberName="";
```

[Table of Contents] [Description]

Path 182:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
121. public void addForum(String lastPostMemberName, String
forumName, //forumxml.java
      ...
      151. if (forumDesc==null) forumDesc="";
```

[Table of Contents] [Description]

Path 183:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
121. public void addForum(String lastPostMemberName, String
forumName, //forumxml.java
      ...
      161. if (forumPassword==null) forumPassword="";
```

[Table of Contents] [Description]

Path 184:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
346. public void addForumWatch(String memberName, //forumxml.java
      ...
      365. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 185:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
96. public void addGroup(String groupOwnerName, String groupName, //groupxml.java
      ...
102. if (groupID>=0) strGroupID=Integer.toString(groupID);
```

[Table of Contents] [Description]

Path 186:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
      ...
142. if (groupOwnerName==null) groupOwnerName="";
```

[Table of Contents] [Description]

Path 187:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
      ...
143. if (groupDesc==null) groupDesc="";
```

[Table of Contents] [Description]

Path 188:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
129. public void addGroup(String strGroupID, //groupxml.java
      ...
154. if (Integer.parseInt(strGroupID)<0) strGroupID=null;
```

[Table of Contents] [Description]

Path 189:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
```

```
213. if (clearInt==0) clearIfError=false;
```

[Table of Contents] [Description]

Path 190:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
434. public static void addFinalErrorHandling(HttpServletRequest request, boolean
resetAllData) //importwebhandler.java
    ...
437. if (resetAllData) message+="Trying to clear the database and create
\"admin\" member (password will be \"admin\")...<br/>";
```

[Table of Contents] [Description]

Path 191:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
643. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 192:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
644. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 193:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
    ...
712. if (sort.length() == 0) sort = "MemberExpireDate";
```

[Table of Contents] [Description]

Path 194:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
     ...
     713. if (order.length()== 0) order = "ASC";
```

[Table of Contents] [Description]

Path 195:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
738. public void prepareListPendingUsers(GenericRequest
request) //memberwebhandler.java
     ...
     750. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 196:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
738. public void prepareListPendingUsers(GenericRequest
request) //memberwebhandler.java
     ...
     751. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 197:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
     844. if (checkList == null) return;
```

[Table of Contents] [Description]

Path 198:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
139. public void addMember(String memberName, String
memberPassword, //memberxml.java
     ...
```

```
168. if (memberID>=0) strMemberID=Integer.toString(memberID);
```

[Table of Contents] [Description]

Path 199:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
     ...
313. if (memberFirstIP==null) memberFirstIP="0.0.0.0";
```

[Table of Contents] [Description]

Path 200:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
     ...
314. if (memberLastIP==null) memberLastIP="0.0.0.0";
```

[Table of Contents] [Description]

Path 201:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
     ...
324. if (memberActivateCode==null) memberActivateCode="";
```

[Table of Contents] [Description]

Path 202:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
     ...
325. if (memberTempPassword==null) memberTempPassword="";
```

[Table of Contents] [Description]

Path 203:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
```

```
...
333. if (memberTitle==null) memberTitle="";
```

[Table of Contents] [Description]

Path 204:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
335. if (memberSignature==null) memberSignature="";
```

[Table of Contents] [Description]

Path 205:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
336. if (memberAvatar==null) memberAvatar="";
```

[Table of Contents] [Description]

Path 206:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
337. if (memberSkin==null) memberSkin="";
```

[Table of Contents] [Description]

Path 207:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
338. if (memberLanguage==null) memberLanguage="";
```

[Table of Contents] [Description]

Path 208:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
```

```
...
341. if (memberAddress==null) memberAddress="";
```

[Table of Contents] [Description]

Path 209:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
342. if (memberCity==null) memberCity="";
```

[Table of Contents] [Description]

Path 210:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
343. if (memberState==null) memberState="";
```

[Table of Contents] [Description]

Path 211:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
344. if (memberCountry==null) memberCountry="";
```

[Table of Contents] [Description]

Path 212:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
345. if (memberPhone==null) memberPhone="";
```

[Table of Contents] [Description]

Path 213:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
```

```
...
346. if (memberMobile==null) memberMobile="";
```

[Table of Contents] [Description]

Path 214:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
347. if (memberFax==null) memberFax="";
```

[Table of Contents] [Description]

Path 215:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
348. if (memberCareer==null) memberCareer="";
```

[Table of Contents] [Description]

Path 216:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
349. if (memberHomepage==null) memberHomepage="";
```

[Table of Contents] [Description]

Path 217:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
670. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 218:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
```

```
...
351. if (memberAol==null) memberAol="";
```

[Table of Contents] [Description]

Path 219:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
352. if (memberIcq==null) memberIcq="";
```

[Table of Contents] [Description]

Path 220:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
353. if (memberMsn==null) memberMsn="";
```

[Table of Contents] [Description]

Path 221:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
354. if (memberCoolLink1==null) memberCoolLink1="";
```

[Table of Contents] [Description]

Path 222:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
...
355. if (memberCoolLink2==null) memberCoolLink2="";
```

[Table of Contents] [Description]

Path 223:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public void addMember(String strMemberID, //memberxml.java
```

```
...
363. if (Integer.parseInt(strMemberID)<0) strMemberID=null;
```

[Table of Contents] [Description]

Path 224:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
110. public void setParentPostID(int value) //postxml.java
     111. if (value<0) parentPostID=-1;
```

[Table of Contents] [Description]

Path 225:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public void setParentThreadID(int value) //postxml.java
     132. if (value<0) parentThreadID=-1;
```

[Table of Contents] [Description]

Path 226:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
147. public void setParentForumID(int value) //postxml.java
     148. if (value<0) parentForumID=-1;
```

[Table of Contents] [Description]

Path 227:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
163. public void setParentCategoryID(int value) //postxml.java
     164. if (value<0) parentCategoryID=-1;
```

[Table of Contents] [Description]

Path 228:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
     ...
     224. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 229:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
     ...
225. if (lastEditMemberName==null) lastEditMemberName="";
```

[Table of Contents] [Description]

Path 230:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
     ...
228. if (postCreationIP==null) postCreationIP="0.0.0.0";
```

[Table of Contents] [Description]

Path 231:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
     ...
229. if (postLastEditIP==null) postLastEditIP="0.0.0.0";
```

[Table of Contents] [Description]

Path 232:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void addPost(String memberName, String
lastEditMemberName, //postxml.java
     ...
234. if (postIcon==null) postIcon="";
```

[Table of Contents] [Description]

Path 233:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
95. public void addRank(String rankMinPosts, String rankLevel, //rankxml.java
      |...
      |112. if (rankImage==null) rankImage="";
```

[Table of Contents] [Description]

Path 234:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
97. public void setParentForumID(int value) //threadxml.java
      |98. if (value<0) parentForumID=-1;
```

[Table of Contents] [Description]

Path 235:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
111. public void setParentCategoryID(int value) //threadxml.java
      |112. if (value<0) parentCategoryID=-1;
```

[Table of Contents] [Description]

Path 236:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
144. public void addThread(String memberName, String
lastPostMemberName, //threadxml.java
      |...
      |177. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 237:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
144. public void addThread(String memberName, String
lastPostMemberName, //threadxml.java
      |...
      |178. if (lastPostMemberName==null) lastPostMemberName="";
```

[Table of Contents] [Description]

Path 238:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
144. public void addThread(String memberName, String
lastPostMemberName, //threadxml.java
      ...
191. if (threadIcon==null) threadIcon="";
```

[Table of Contents] [Description]

Path 239:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
232. public void addThreadWatch(String memberName, //threadxml.java
      ...
251. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 240:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
273. public void addFavoriteThread(String memberName, //threadxml.java
      ...
291. if (memberName==null) memberName="";
```

[Table of Contents] [Description]

Path 241:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
72. public static int stringToIntDef(String value, int defaultValue) //xmlutil.java
      ...
74. if (value==null) return defaultValue;
```

[Table of Contents] [Description]

Path 242:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
87. public static int stringToGenderDef(String value, int
defaultValue) //xmlutil.java
      88. if (value==null) return defaultValue;
```

[Table of Contents] [Description]

Path 243:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
92. public static String genderToString(int gender) //xmlutil.java
      93. if (gender==0) return "0"; //or, is it better to return "female" ?
```

[Table of Contents] [Description]

Path 244:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
92. public static String genderToString(int gender) //xmlutil.java
      ...
      94. else if (gender==1) return "1"; //"male"
```

[Table of Contents] [Description]

Path 245:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
108. public static boolean stringToBooleanDef(String value, boolean
defaultValue) //xmlutil.java
      109. if (value==null) return defaultValue;
```

[Table of Contents] [Description]

Path 246:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
113. public static String booleanToString(boolean value) //xmlutil.java
      114. if (value) return "true";
```

[Table of Contents] [Description]

Path 247:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
118. public static java.sql.Date stringToSqlDate(String s) //xmlutil.java
      ...
      125. if (s==null) throw new java.lang.IllegalArgumentException("null string");
```

[Table of Contents] [Description]

Path 248:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
118. public static java.sql.Date stringToSqlDate(String s) //xmlutil.java
```

```
...
135. if (s.indexOf('/')>0) f=/*f1*/new SimpleDateFormat("yyyy/MM/dd",
Locale.US);
```

[Table of Contents] [Description]

Path 249:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
118. public static java.sql.Date stringToSqlDate(String s) //xmlutil.java
...
136. else if (s.indexOf('-')>0) f=/*f2*/new SimpleDateFormat("yyyy-MM-dd",
Locale.US);
```

[Table of Contents] [Description]

Path 250:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
118. public static java.sql.Date stringToSqlDate(String s) //xmlutil.java
...
137. else if (s.indexOf(' ')>0) f=/*f4*/new SimpleDateFormat("EEE MMM dd
yyyy", Locale.US);
```

[Table of Contents] [Description]

Path 251:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
145. public static java.sql.Date stringToSqlDateDef(String value, java.sql.Date
defaultValue) //xmlutil.java
146. if (value==null) return defaultValue;
```

[Table of Contents] [Description]

Path 252:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
173. public static String sqlDateToStringDefNow(java.sql.Date value) //xmlutil.java
...
175. if (value==null) return sqlDateToString(new java.sql.Date(now.getTime
()));
```

[Table of Contents] [Description]

Path 253:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
179. public static String sqlDateToStringDefEmpty(java.sql.Date
value) //xmlutil.java
     ...
181. if (value==null) return "";
```

[Table of Contents] [Description]

Path 254:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
185. public static java.sql.Timestamp stringToSqlTimestamp(String s) //xmlutil.java
     ...
193. if (s==null) throw new java.lang.IllegalArgumentException("null string");
```

[Table of Contents] [Description]

Path 255:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
224. public static java.sql.Timestamp stringToSqlTimestampDef(String value,
java.sql.Timestamp defaultValue) //xmlutil.java
225. if (value==null) return defaultValue;
```

[Table of Contents] [Description]

Path 256:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
252. public static String sqlTimestampToStringDefNow(java.sql.Timestamp
value) //xmlutil.java
     ...
254. if (value==null) return sqlTimestampToString(now);
```

[Table of Contents] [Description]

Path 257:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
258. public static String sqlTimestampToStringDefEmpty(java.sql.Timestamp
value) //xmlutil.java
259. if (value==null) return "";
```

[Table of Contents] [Description]

Path 258:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
125. public void close() throws IOException //xmlwriter.java
     ...
127. if (outWriter!=null) outWriter.close();
```

[Table of Contents] [Description]

Path 259:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
125. public void close() throws IOException //xmlwriter.java
     ...
127. if (outWriter!=null) outWriter.close();
```

[Table of Contents] [Description]

Path 260:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
148. public void startElement(String elementName, String[]
attrNameValue) //xmlwriter.java
     ...
153. if (attrNameValue!=null) for (int i=0; i<attrNameValue.length-1; i+=2) {
```

[Table of Contents] [Description]

Path 261:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
165. public void endElement(String elementName) //xmlwriter.java
     ...
169. if (!startedNewElement) outputNewLine();
```

[Table of Contents] [Description]

Path 262:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
165. public void endElement(String elementName) //xmlwriter.java
     ...
169. if (!startedNewElement) outputNewLine();
```

[Table of Contents] [Description]

Path 263:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
180. public void writeComment(String comment) throws IOException //xmlwriter.java
       ...
182. /*if (startedNewElement)*/ outputNewLine();
```

[Table of Contents] [Description]

Path 264:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
197. public void encodeAndWriteData(String data) throws IOException //xmlwriter.java
       ...
202. if ((amp>=0) && (lt>=0)) i=(amp<lt)?amp:lt;
```

[Table of Contents] [Description]

Path 265:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
197. public void encodeAndWriteData(String data) throws IOException //xmlwriter.java
       ...
203. else if (amp>=0) i=amp;
```

[Table of Contents] [Description]

Path 266:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
197. public void encodeAndWriteData(String data) throws IOException //xmlwriter.java
       ...
223. if ((amp>=0) && (lt>=0)) i=(amp<lt)?amp:lt;
```

[Table of Contents] [Description]

Path 267:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
197. public void encodeAndWriteData(String data) throws IOException //xmlwriter.java
       ...
224. else if (amp>=0) i=amp;
```

[Table of Contents] [Description]

Path 268:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
237. private void processBufferedData() throws IOException //xmlwriter.java
     |238. if (textBuffer==null) return;
```

[Table of Contents] [Description]

Path 269:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
237. private void processBufferedData() throws IOException //xmlwriter.java
     |...
     |240. if (!s.trim().equals("")) textBuffer=null;
```

[Table of Contents] [Description]

Path 270:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
237. private void processBufferedData() throws IOException //xmlwriter.java
     |...
     |241. if (s.equals("")) return;
```

[Table of Contents] [Description]

Path 271:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
237. private void processBufferedData() throws IOException //xmlwriter.java
     |...
     |243. if (startedNewElement) outputNewLine();
```

[Table of Contents] [Description]

Path 272:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
237. private void processBufferedData() throws IOException //xmlwriter.java
     |...
     |243. if (startedNewElement) outputNewLine();
```

[Table of Contents] [Description]

Path 273:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     |...
     |116. if (name.equals("ForumPasswords")) JiveXML.allForumsPassword=value;
```

[Table of Contents] [Description]

Path 274:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     |...
     |117. else if (name.equals("RootCategory")) JiveXML.rootCategoryName=value;
```

[Table of Contents] [Description]

Path 275:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     |...
     |118. else if (name.equals("RootCategoryDesc")) JiveXML.rootCategoryDesc=value;
```

[Table of Contents] [Description]

Path 276:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     |...
     |119. else if (name.equals("JiveGuest")) JiveXML.guestName=value;
```

[Table of Contents] [Description]

Path 277:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
106. public static void importXml(File importFile, //importjive.java
     |...
     |120. else if (name.equals("JiveAdmin")) JiveXML.adminName=value;
```

[Table of Contents] [Description]

Path 278:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      |...
      |500. if (e==null) log.error(message);
```

[Table of Contents] [Description]

Path 279:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      |...
      |500. if (e==null) log.error(message);
```

[Table of Contents] [Description]

Path 280:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      |...
      |500. if (e==null) log.error(message);
```

[Table of Contents] [Description]

Path 281:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      |...
      |502. if ((e==null) || (e.getMessage()==null)) addErrorMessage(message);
```

[Table of Contents] [Description]

Path 282:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      |...
```

```
502. if ((e==null) || (e.getMessage()==null)) addErrorMessage(message);
```

[Table of Contents] [Description]

Path 283:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
502. if ((e==null) || (e.getMessage()==null)) addErrorMessage(message);
```

[Table of Contents] [Description]

Path 284:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
507. if (e==null) throw new ImportException(message);
```

[Table of Contents] [Description]

Path 285:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
113. public void addJiveForum() //jiveforumxml.java
...
121. if (forumCreated) return;
```

[Table of Contents] [Description]

Path 286:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
142. public void updateAddedThread(ThreadXML threadXML) throws
ObjectNotFoundException, //jiveforumxml.java
...
144. if ((!forumCreated) || (forumXML.getForumID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 287:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
149. public void updateAddedPost(PostXML postXML, String postUsername, Timestamp
postCreationDate) //jiveforumxml.java
     ...
151. if ((!forumCreated) || (forumXML.getForumID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 288:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
109. public void addJiveGroup() throws CreateException,
DuplicateKeyException, //jivegroupxml.java
     ...
116. if (groupCreated) return;
```

[Table of Contents] [Description]

Path 289:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
152. public void addJiveMessage() throws CreateException,
DuplicateKeyException, //jivemessagexml.java
     ...
159. if (postCreated) return;
```

[Table of Contents] [Description]

Path 290:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
196. public void updateAddedReplyPost(PostXML subPost, String postUsername, String
postCreationDate) //jivemessagexml.java
     ...
198. if ((!postCreated) || (postXML.getPostID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 291:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
137. public void addJiveThread() throws BadInputException,
CreateException, //jivethreadxml.java
     ...
```

```
145. if (!threadCreated) addJiveThread(threadUsername, threadTopic,
threadBody);
```

[Table of Contents] [Description]

Path 292:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
137. public void addJiveThread() throws BadInputException,
CreateException, //jivethreadxml.java
    ...
145. if (!threadCreated) addJiveThread(threadUsername, threadTopic,
threadBody);
```

[Table of Contents] [Description]

Path 293:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
148. public void addJiveThread(String firstMessageUsername, String
firstMessageSubject, String firstMessageBody) //jivethreadxml.java
    ...
156. if (threadCreated) return;
```

[Table of Contents] [Description]

Path 294:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
85. public void addJiveUser(String username, String password, String
email, //jiveuserxml.java
    ...
97. if (memberCreated) return;
```

[Table of Contents] [Description]

Path 295:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
85. public void addJiveUser(String username, String password, String
email, //jiveuserxml.java
    ...
116. if (emailVisible==null) emailVisible="true";
```

[Table of Contents] [Description]

Path 296:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
85. public void addJiveUser(String username, String password, String
email, //jiveuserxml.java
      ...
117. if (nameVisible==null) nameVisible="true";
```

[Table of Contents] [Description]

Path 297:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
236. if ((importFile!=null) && (importFile.exists())) importFile.delete();
```

[Table of Contents] [Description]

Path 298:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
236. if ((importFile!=null) && (importFile.exists())) importFile.delete();
```

[Table of Contents] [Description]

Path 299:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
237. if ((importXml!=null) && (importXml.exists())) importXml.delete();
```

[Table of Contents] [Description]

Path 300:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
237. if ((importXml!=null) && (importXml.exists())) importXml.delete();
```

[Table of Contents] [Description]

Path 301:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
262. if (!avatarsDir.endsWith(File.separator)) avatarsDir+=File.separator;
```

[Table of Contents] [Description]

Path 302:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
263. if (!attachsDir.endsWith(File.separator)) attachsDir+=File.separator;
```

[Table of Contents] [Description]

Path 303:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
264. if (!mvnForumHomeDir.endsWith(File.separator))
mvnForumHomeDir+=File.separator;
```

[Table of Contents] [Description]

Path 304:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
     ...
271. if (zipEntry==null) break;
```

[Table of Contents] [Description]

Path 305:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
```

```
request) //importmvnforum.java
    ...
281. if (thisAvatar.length()<=0) destPath=null;
```

[Table of Contents] [Description]

Path 306:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
245. private static File extractImportZip(File importZipFile, HttpServletRequest
request) //importmvnforum.java
    ...
292. if (thisAttach.length()<=0) destPath=null;
```

[Table of Contents] [Description]

Path 307:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
85. public void setAttachmentID(String id) //mvnforumattachmentxml.java
    86. if (id!=null) attachmentXML.setAttachmentID(id);
```

[Table of Contents] [Description]

Path 308:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
85. public void setAttachmentID(String id) //mvnforumattachmentxml.java
    86. if (id!=null) attachmentXML.setAttachmentID(id);
```

[Table of Contents] [Description]

Path 309:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
169. public void addAttachment() throws CreateException,
DuplicateKeyException, //mvnforumattachmentxml.java
    ...
176. if (attachmentCreated) return;
```

[Table of Contents] [Description]

Path 310:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
142. public void addCategory() throws CreateException,
DuplicateKeyException, //mvnforumcategoryxml.java
    ...
149. if (categoryCreated) return;
```

[Table of Contents] [Description]

Path 311:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
182. public void addForum() throws CreateException,
DuplicateKeyException, //mvnforumforumxml.java
    ...
189. if (forumCreated) return;
```

[Table of Contents] [Description]

Path 312:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
244. public void updateAddedThread(MvnForumThreadXML
subThread) //mvnforumforumxml.java
    245. if ((!forumCreated) || (forumXML.getForumID()<0)) return; //todo Igor:
    process this error
```

[Table of Contents] [Description]

Path 313:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
253. public void updateAddedPost(MvnForumPostXML subPost, String postUsername,
String postCreationDate) //mvnforumforumxml.java
    254. if ((!forumCreated) || (forumXML.getForumID()<0)) return; //todo Igor:
    process this error
```

[Table of Contents] [Description]

Path 314:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
264. public void updateAddedAttachment(MvnForumAttachmentXML
subAttachment) //mvnforumforumxml.java
    265. if ((!forumCreated) || (forumXML.getForumID()<0)) return; //todo Igor:
    process this error
```

[Table of Contents] [Description]

Path 315:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public void addGroup() throws CreateException,
DuplicateKeyException, //mvnforumgroupxml.java
      |
      ...
      138. if (groupCreated) return;
```

[Table of Contents] [Description]

Path 316:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public void addGroup() throws CreateException,
DuplicateKeyException, //mvnforumgroupxml.java
      |
      ...
      144. if (!isRegisteredMembersGroup) MvnForumXML.checkRegisteredMembersGroup();
```

[Table of Contents] [Description]

Path 317:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public void addGroup() throws CreateException,
DuplicateKeyException, //mvnforumgroupxml.java
      |
      ...
      144. if (!isRegisteredMembersGroup) MvnForumXML.checkRegisteredMembersGroup();
```

[Table of Contents] [Description]

Path 318:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
131. public void addGroup() throws CreateException,
DuplicateKeyException, //mvnforumgroupxml.java
      |
      ...
      150. if (isRegisteredMembersGroup)
      MvnForumXML.addedRegisteredMembersGroup=true;
```

[Table of Contents] [Description]

Path 319:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
      |
```

```
...
366. if (memberCreated) return;
```

[Table of Contents] [Description]

Path 320:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
...
373. if (!isAdmin) MvnForumXML.checkAdminMember();
```

[Table of Contents] [Description]

Path 321:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
...
373. if (!isAdmin) MvnForumXML.checkAdminMember();
```

[Table of Contents] [Description]

Path 322:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
...
374. if ((!isAdmin) && (!isGuest)) MvnForumXML.checkGuestMember();
```

[Table of Contents] [Description]

Path 323:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
...
374. if ((!isAdmin) && (!isGuest)) MvnForumXML.checkGuestMember();
```

[Table of Contents] [Description]

Path 324:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
```

```
...
395. if (isAdmin) MvnForumXML.addedAdminMember=true;
```

[Table of Contents] [Description]

Path 325:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
357. public void addMember() //mvnforummemberxml.java
...
396. else if (isGuest) MvnForumXML.addedGuestMember=true;
```

[Table of Contents] [Description]

Path 326:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
192. public void addPost() throws CreateException,
DuplicateKeyException, //mvnforumpostxml.java
...
199. if (postCreated) return;
```

[Table of Contents] [Description]

Path 327:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
232. public void updateAddedReply(MvnForumPostXML subPost, String postUsername,
String postCreationDate) //mvnforumpostxml.java
233. if ((!postCreated) || (postXML.getPostID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 328:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
247. public void updateAddedAttachment(MvnForumAttachmentXML
subAttachment) //mvnforumpostxml.java
248. if ((!postCreated) || (postXML.getPostID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 329:

Query Name - **Single_Line_If_Statement**

Severity

- **Information**

```
201. public void addThread() throws CreateException,
DuplicateKeyException, //mvnforumthreadxml.java
...
208. if (threadCreated) return;
```

[Table of Contents] [Description]

Path 330:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public void updateAddedPost(MvnForumPostXML subPost, String postUsername,
String postCreationDate) //mvnforumthreadxml.java
262. if ((!threadCreated) || (threadXML.getThreadID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 331:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
274. public void updateAddedAttachment(MvnForumAttachmentXML
subAttachment) //mvnforumthreadxml.java
275. if ((!threadCreated) || (threadXML.getThreadID()<0)) return; //todo Igor:
process this error
```

[Table of Contents] [Description]

Path 332:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
215. public boolean updateNewMessageCount(boolean forceUpdate) //onlineuserimpl.java
...
217. if (isGuest()) return false;
```

[Table of Contents] [Description]

Path 333:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
257. public synchronized String getGMTDateFormat(java.util.Date gmtDate, boolean
adjustTimeZone) //onlineuserimpl.java
258. if (gmtDate == null) return "";
```

[Table of Contents] [Description]

Path 334:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
271. public synchronized String getGMTTimestampFormat(Timestamp gmtTimestamp,
boolean adjustTimeZone) //onlineuserimpl.java
     272. if (gmtTimestamp == null) return "";
```

[Table of Contents] [Description]

Path 335:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
142. public String drawCategory(CategoryTreeEvent event) //modcpimpl.java
     ...
     158. if (MyUtil.canViewAnyForumInCategory(categoryBean.getCategoryID(),
     permission) == false)
```

[Table of Contents] [Description]

Path 336:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
146. public String drawCategory(CategoryTreeEvent event) //rssimpl.java
     ...
     158. if (MyUtil.canViewAnyForumInCategory(categoryBean.getCategoryID(),
     permission) == false)
```

[Table of Contents] [Description]

Path 337:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
496. public String getMemberGenderString() //memberbean.java
     497. if (memberGender == MEMBER_GENDER_MALE) return "Male";
```

[Table of Contents] [Description]

Path 338:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
522. public String getMemberHomepage_http() //memberbean.java
     523. if (memberHomepage == null) memberHomepage = "";
```

[Table of Contents] [Description]

Path 339:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
534. public String getMemberCoolLink1_http() //memberbean.java
     |535. if (memberCoolLink1 == null) memberCoolLink1 = "";
```

[Table of Contents] [Description]

Path 340:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
546. public String getMemberCoolLink2_http() //memberbean.java
     |547. if (memberCoolLink2 == null) memberCoolLink2 = "";
```

[Table of Contents] [Description]

Path 341:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
697. public Collection getAttachments_withSortSupport_limit(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
     |...
     |700. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
     allowed.");
```

[Table of Contents] [Description]

Path 342:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
697. public Collection getAttachments_withSortSupport_limit(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
     |...
     |701. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
     rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 343:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
788. public Collection getAttachments_withSortSupport_limit_general(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
     |...
     |848. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 344:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
887. public Collection getAttachments_fromIDRange(int fromID, int
toID) //attachmentdaoimpljdbc.java
      ...
      890. if (fromID < 0) throw new IllegalArgumentException("The fromID < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 345:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
887. public Collection getAttachments_fromIDRange(int fromID, int
toID) //attachmentdaoimpljdbc.java
      ...
      891. if (toID < fromID) throw new IllegalArgumentException("toID < fromID is
      not allowed.");
```

[Table of Contents] [Description]

Path 346:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      920. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 347:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      921. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 348:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      978. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 349:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      997. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 350:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      998. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 351:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1041. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 352:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
```

```
DatabaseException //forumdaoimpljdbc.java
    ...
1062. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 353:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
    1080. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
    not allowed.");
```

[Table of Contents] [Description]

Path 354:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
    ...
    1081. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
    rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 355:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
    ...
    1121. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 356:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
    ...
    1142. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 357:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1182. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
      not allowed.");
```

[Table of Contents] [Description]

Path 358:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1183. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 359:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1247. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 360:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1264. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
      not allowed.");
```

[Table of Contents] [Description]

Path 361:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
1265. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 362:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
1329. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 363:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
      ...
911. if (retValue == null) retValue = ""; // hack for Oracle database
```

[Table of Contents] [Description]

Path 364:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
      ...
1006. if (retValue == null) retValue = ""; // hack for Oracle database
```

[Table of Contents] [Description]

Path 365:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      |
```

```
...
1498. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 366:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1499. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 367:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1629. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 368:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1630. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 369:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1682. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 370:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
1727. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 371:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
1763. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 372:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
1764. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 373:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
1865. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 374:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
    ...
2132. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 375:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
    ...
2133. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 376:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
    ...
2189. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 377:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2370. public Collection getMembers_fromIDRange(int fromID, int toID) throws
IllegalArgumentException, DatabaseException //memberdaoimpljdbc.java
    ...
2372. if (fromID < 0) throw new IllegalArgumentException("The fromID < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 378:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2370. public Collection getMembers_fromIDRange(int fromID, int toID) throws
IllegalArgumentException, DatabaseException //memberdaoimpljdbc.java
    ...
2373. if (toID < fromID) throw new IllegalArgumentException("toID < fromID is
```

```
not allowed.");
```

[Table of Contents] [Description]

Path 379:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
326. public Collection getBeans_limit(int groupID, int offset, int
rowsToReturn) //membergroupdaoimpljdbc.java
    ...
    328. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
    allowed.");
```

[Table of Contents] [Description]

Path 380:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
326. public Collection getBeans_limit(int groupID, int offset, int
rowsToReturn) //membergroupdaoimpljdbc.java
    ...
    329. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
    rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 381:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
234. public Collection getAllMessages_inMember_inFolder_withSortSupport_limit(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order) //messagedaoimpljdbc.java
    ...
    237. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
    allowed.");
```

[Table of Contents] [Description]

Path 382:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
234. public Collection getAllMessages_inMember_inFolder_withSortSupport_limit(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order) //messagedaoimpljdbc.java
    ...
    238. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
    rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 383:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public Collection getNonPublicMessages_inMember_inFolder_withSortSupport_limit
(int memberID, String folderName, int offset, int rowsToReturn, String sort, String
order) //messagedaoimpljdbc.java
      ...
      264. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 384:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public Collection getNonPublicMessages_inMember_inFolder_withSortSupport_limit
(int memberID, String folderName, int offset, int rowsToReturn, String sort, String
order) //messagedaoimpljdbc.java
      ...
      265. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 385:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
364. public Collection getBeans_inMember_inFolder_withSortSupport_limit_noscroll(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      ...
      393. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 386:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
364. public Collection getBeans_inMember_inFolder_withSortSupport_limit_noscroll(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      ...
      415. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 387:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
436. public Collection getBeans_inMember_inFolder_withSortSupport_limit_general(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
        ...
490. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 388:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
683. private Collection getBeans_inThread_limit_mysql(int threadID, int offset, int
rowsToReturn, boolean enable) //postdaoimpljdbc.java
        ...
685. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
allowed.");
```

[Table of Contents] [Description]

Path 389:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
683. private Collection getBeans_inThread_limit_mysql(int threadID, int offset, int
rowsToReturn, boolean enable) //postdaoimpljdbc.java
        ...
686. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 390:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
751. private Collection getBeans_inThread_limit_noscroll(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
        ...
753. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
allowed.");
```

[Table of Contents] [Description]

Path 391:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
751. private Collection getBeans_inThread_limit_noscroll(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
    ...
754. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 392:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
751. private Collection getBeans_inThread_limit_noscroll(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
    ...
779. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 393:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
751. private Collection getBeans_inThread_limit_noscroll(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
    ...
801. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 394:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
    ...
824. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
allowed.");
```

[Table of Contents] [Description]

Path 395:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
    ...
825. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 396:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
      ...
      876. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 397:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1078. public Collection getLastEnablePosts_inThread_limit(int threadID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
      1080. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 398:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1078. public Collection getLastEnablePosts_inThread_limit(int threadID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
      1125. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 399:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1146. public Collection getLastEnablePosts_inForum_limit(int forumID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
      1148. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 400:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1146. public Collection getLastEnablePosts_inForum_limit(int forumID, int
rowsToReturn) //postdaoimpljdbc.java
      ...
1193. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 401:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1312. public Collection getPosts_fromIDRange(int fromID, int
toID) //postdaoimpljdbc.java
      ...
1315. if (fromID < 0) throw new IllegalArgumentException("The fromID < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 402:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1312. public Collection getPosts_fromIDRange(int fromID, int
toID) //postdaoimpljdbc.java
      ...
1316. if (toID < fromID) throw new IllegalArgumentException("toID < fromID is
not allowed.");
```

[Table of Contents] [Description]

Path 403:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1373. public Collection getMostActiveMembers(Timestamp since, int
rowsToReturn) //postdaoimpljdbc.java
      ...
1402. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 404:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1416. public Collection getMostActiveThreads(Timestamp since, int
rowsToReturn) //postdaoimpljdbc.java
      ...
```

```
1460. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 405:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
798. public Collection getEnableThreads_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
     ...
801. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
allowed.");
```

[Table of Contents] [Description]

Path 406:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
798. public Collection getEnableThreads_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
     ...
802. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 407:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
845. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
     ...
848. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
allowed.");
```

[Table of Contents] [Description]

Path 408:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
845. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
```

```
...
849. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 409:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1018. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
...
1021. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 410:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1018. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
...
1022. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 411:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1018. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
...
1095. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 412:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1018. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
```

```
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
    ...
1118. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 413:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1132. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
    ...
1158. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 414:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1132. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
    ...
1173. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 415:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
    ...
1197. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 416:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
```

```
forum) //threaddaoimpljdbc.java
        ...
1198. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 417:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
        ...
1298. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 418:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1313. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
        ...
1356. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 419:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1414. private Collection getBeans_inForum_withSortSupport_limit_mysql(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
        ...
1417. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 420:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1414. private Collection getBeans_inForum_withSortSupport_limit_mysql(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
```

```
...
1418. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 421:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1504. private Collection getBeans_inForum_withSortSupport_limit_noscroll(int
forumID, int offset, int rowsToReturn, String sort, String order, boolean enable,
boolean onlyNormalThread) //threaddaoimpljdbc.java
...
1507. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 422:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1504. private Collection getBeans_inForum_withSortSupport_limit_noscroll(int
forumID, int offset, int rowsToReturn, String sort, String order, boolean enable,
boolean onlyNormalThread) //threaddaoimpljdbc.java
...
1508. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 423:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1504. private Collection getBeans_inForum_withSortSupport_limit_noscroll(int
forumID, int offset, int rowsToReturn, String sort, String order, boolean enable,
boolean onlyNormalThread) //threaddaoimpljdbc.java
...
1550. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 424:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1504. private Collection getBeans_inForum_withSortSupport_limit_noscroll(int
forumID, int offset, int rowsToReturn, String sort, String order, boolean enable,
boolean onlyNormalThread) //threaddaoimpljdbc.java
```

```
...
1573. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 425:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1600. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 426:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1601. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 427:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1669. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 428:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2275. private Collection
getEnableThreadsWithPendingPosts_withSortSupport_limit_noscroll(int forumID, int
offset, int rowsToReturn, String sort, String order) //threaddaoimpljdbc.java
...
```

```
2278. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is
not allowed.");
```

[Table of Contents] [Description]

Path 429:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2275. private Collection
getEnableThreadsWithPendingPosts_withSortSupport_limit_noscroll(int forumID, int
offset, int rowsToReturn, String sort, String order) //threaddaoimpljdbc.java
    ...
2279. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 430:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2275. private Collection
getEnableThreadsWithPendingPosts_withSortSupport_limit_noscroll(int forumID, int
offset, int rowsToReturn, String sort, String order) //threaddaoimpljdbc.java
    ...
2326. if (rowIndex < offset) continue;
```

[Table of Contents] [Description]

Path 431:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
2275. private Collection
getEnableThreadsWithPendingPosts_withSortSupport_limit_noscroll(int forumID, int
offset, int rowsToReturn, String sort, String order) //threaddaoimpljdbc.java
    ...
2337. if (retValue.size() == rowsToReturn) break;// Fix the Sybase bug
```

[Table of Contents] [Description]

Path 432:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
185. static void doIndexAttachment(AttachmentBean attachmentBean, IndexWriter
writer) throws SearchException //attachmentindexer.java
    ...
187. if (attachmentBean == null) return;
```

[Table of Contents] [Description]

Path 433:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
239. private Collection getAttachments(Hits attachHits, int offset, int
rowsToReturn) //attachmentsearchquery.java
      ...
      242. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 434:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
239. private Collection getAttachments(Hits attachHits, int offset, int
rowsToReturn) //attachmentsearchquery.java
      ...
      243. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
      rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 435:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
186. static void doIndexMember(MemberBean memberBean, IndexWriter writer) throws
SearchException //memberindexer.java
      187. if (memberBean == null) return;
```

[Table of Contents] [Description]

Path 436:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. private Collection getMembers(Hits MemberHits, int offset, int
rowsToReturn) //membersearchquery.java
      ...
      252. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
      allowed.");
```

[Table of Contents] [Description]

Path 437:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. private Collection getMembers(Hits MemberHits, int offset, int
rowsToReturn) //membersearchquery.java
     ...
     253. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
     rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 438:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
198. static void doIndexPost(PostBean post, IndexWriter writer) throws
SearchException //postindexer.java
     ...
     200. if (post == null) return;
```

[Table of Contents] [Description]

Path 439:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
291. private Collection getPosts(Hits postHits, int offset, int
rowsToReturn) //postsearchquery.java
     ...
     294. if (offset < 0) throw new IllegalArgumentException("The offset < 0 is not
     allowed.");
```

[Table of Contents] [Description]

Path 440:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
291. private Collection getPosts(Hits postHits, int offset, int
rowsToReturn) //postsearchquery.java
     ...
     295. if (rowsToReturn <= 0) throw new IllegalArgumentException("The
     rowsToReturn <= 0 is not allowed.");
```

[Table of Contents] [Description]

Path 441:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
     ...
     87. if (posts == null) return;
```

[Table of Contents] [Description]

Path 442:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1174. public void prepareListMembers_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1198. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 443:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1174. public void prepareListMembers_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1199. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 444:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
84. public void processAdd(GenericRequest request) //messagefolderwebhandler.java
      ...
112. if (folderOrder < 10) folderOrder = 10;
```

[Table of Contents] [Description]

Path 445:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
514. public void prepareList(GenericRequest request) //messagewebhandler.java
      ...
544. if (sort.length() == 0) sort = "MessageCreationDate";
```

[Table of Contents] [Description]

Path 446:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
514. public void prepareList(GenericRequest request) //messagewebhandler.java
      ...
```

```
545. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 447:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
...
849. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 448:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
...
850. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 449:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
920. public void prepareListRecentThreads_limit(GenericRequest
request,GenericResponse response) //threadwebhandler.java
...
932. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 450:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
920. public void prepareListRecentThreads_limit(GenericRequest
request,GenericResponse response) //threadwebhandler.java
...
933. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 451:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
973. public void prepareListRecentDisabledThreads_limit(GenericRequest
request) //threadwebhandler.java
    ...
988. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 452:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
973. public void prepareListRecentDisabledThreads_limit(GenericRequest
request) //threadwebhandler.java
    ...
989. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 453:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1057. public void prepareListRecentEnableThreadsWithPendingPosts_limit
(GenericRequest request) //threadwebhandler.java
    ...
1072. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 454:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1057. public void prepareListRecentEnableThreadsWithPendingPosts_limit
(GenericRequest request) //threadwebhandler.java
    ...
1073. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 455:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1107. public void prepareListEnableThreadsWithPendingPosts_inForum_limit
(GenericRequest request) //threadwebhandler.java
    ...
1126. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 456:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1107. public void prepareListEnableThreadsWithPendingPosts_inForum_limit
(GenericRequest request) //threadwebhandler.java
      ...
1127. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 457:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1197. public void prepareModeratePendingThreads_inForum_limit(GenericRequest
request) //threadwebhandler.java
      ...
1219. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 458:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1197. public void prepareModeratePendingThreads_inForum_limit(GenericRequest
request) //threadwebhandler.java
      ...
1220. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 459:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1399. public void prepareRSSSummary(GenericRequest request, GenericResponse
response) //threadwebhandler.java
      ...
1405. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 460:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1399. public void prepareRSSSummary(GenericRequest request, GenericResponse
response) //threadwebhandler.java
      ...
1406. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 461:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1427. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 462:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1428. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 463:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1434. if (rows <= 0) rows = MVNForumConfig.getRowsPerRSS();
```

[Table of Contents] [Description]

Path 464:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1487. public void prepareListUnansweredThreads(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
1498. if (sort.length() == 0) sort = "ThreadCreationDate";
```

[Table of Contents] [Description]

Path 465:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1487. public void prepareListUnansweredThreads(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
1499. if (order.length()== 0) order = "DESC";
```

[Table of Contents] [Description]

Path 466:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //inc_common.jsp
      ...
81. if (currentLocale.equals("")) currentLocale =
MVNForumConfig.getDefaultLocaleName();
```

[Table of Contents] [Description]

Path 467:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listlogfiles.jsp
      ...
89. if (currentLogFile.equals(logFiles[i])) isCurrent = true;
```

[Table of Contents] [Description]

Path 468:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listpendingusers.jsp
      ...
92. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 469:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listpendingusers.jsp
      ...
93. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 470:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listuserexpire.jsp
      ...
      108. if (sort.length() == 0) sort = "MemberExpireDate";
```

[Table of Contents] [Description]

Path 471:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listuserexpire.jsp
      ...
      109. if (order.length() == 0) order = "ASC";
```

[Table of Contents] [Description]

Path 472:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listuserexpire.jsp
      ...
      209. if (memberExpireDate == null) memberExpireDate = now;
```

[Table of Contents] [Description]

Path 473:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //logframe.jsp
      ...
      69. if (lines[i] == null) break;// all last elements is null
```

[Table of Contents] [Description]

Path 474:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //logframe.jsp
      ...
      70. if ( (log.equals("ALL") == false) && (lines[i].indexOf(log) < 0) )
      continue; %>
```

[Table of Contents] [Description]

Path 475:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //searchmember.jsp
       ...
196. if (sort.length() == 0) sort = "MemberName";
```

[Table of Contents] [Description]

Path 476:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //searchmember.jsp
       ...
197. if (order.length() == 0) order = "ASC";
```

[Table of Contents] [Description]

Path 477:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //usermanagement.jsp
       ...
149. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 478:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //usermanagement.jsp
       ...
150. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 479:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //viewlogsystem.jsp
       ...
159. if (lines[i] == null ) break;// all last elements is null
```

[Table of Contents] [Description]

Path 480:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //viewlogsystem.jsp
       ...
160. if ( !log.equals("ALL") && (lines[i].indexOf(log) < 0) ) continue; %>
```

[Table of Contents] [Description]

Path 481:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //addmessageattachmentsuccess.jsp
       ...
55. if (overQuotaReceivers == null) overQuotaReceivers = "";
```

[Table of Contents] [Description]

Path 482:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //addmessagesuccess.jsp
       ...
52. if (overQuotaReceivers == null) overQuotaReceivers = "";
```

[Table of Contents] [Description]

Path 483:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //addpost.jsp
       ...
82. if (action == null) action = "";
```

[Table of Contents] [Description]

Path 484:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //addwatch.jsp
       ...
188. if (MyUtil.canViewAnyForumInCategory(categoryID, permission) == false)
continue; %>
```

[Table of Contents] [Description]

Path 485:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //addwatch.jsp
      ...
      195. if (forumBean.getCategoryID() != categoryID) continue;
```

[Table of Contents] [Description]

Path 486:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //inc_common.jsp
      ...
      82. if (currentLocale.equals("")) currentLocale =
      MVNForumConfig.getDefaultLocaleName();
```

[Table of Contents] [Description]

Path 487:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listmembers.jsp
      ...
      87. if (sort.length() == 0) sort = "MemberCreationDate";
```

[Table of Contents] [Description]

Path 488:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listmembers.jsp
      ...
      88. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 489:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentpendingthreads.jsp
      ...
      92. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 490:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentpendingthreads.jsp
       ...
93. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 491:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentthreads.jsp
       ...
136. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 492:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentthreads.jsp
       ...
137. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 493:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentthreadswithpendingposts.jsp
       ...
93. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 494:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listrecentthreadswithpendingposts.jsp
       ...
94. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 495:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listthreads.jsp
       ...
157. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 496:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listthreads.jsp
       ...
158. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 497:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listthreadswithpendingposts.jsp
       ...
83. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 498:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listthreadswithpendingposts.jsp
       ...
84. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 499:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listunansweredthreads.jsp
       ...
174. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 500:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //listunansweredthreads.jsp
       ...
175. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 501:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //moderatependingthreads.jsp
       ...
117. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 502:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //moderatependingthreads.jsp
       ...
118. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 503:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //mymessage.jsp
       ...
80. if (sort.length() == 0) sort = "MessageCreationDate";
```

[Table of Contents] [Description]

Path 504:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //mymessage.jsp
       ...
81. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 505:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //mymessage.jsp
      ...
      82. if (folder.length() == 0) folder = "Inbox";
```

[Table of Contents] [Description]

Path 506:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //mymessage.jsp
      ...
      321. if (messageBean.getFolderName().equalsIgnoreCase
      (MVNForumConstant.MESSAGE_FOLDER_DRAFT) && (messageBean.getMemberID() !=
      onlineUser.getMemberID())) continue;
```

[Table of Contents] [Description]

Path 507:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //rsssummary.jsp
      ...
      76. if (sort.length() == 0) sort = "ThreadLastPostDate";
```

[Table of Contents] [Description]

Path 508:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //rsssummary.jsp
      ...
      77. if (order.length() == 0) order = "DESC";
```

[Table of Contents] [Description]

Path 509:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
1. <%-- //viewthread.jsp
      ...
      715. if ((postID >=0) && (postBean.getPostID() != postID)) continue;
```

[Table of Contents] [Description]

Path 510:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
352. Connection getConnection(long timeout) //dbconnectionmanager.java
     ...
363. if (timeToWait > TIME_BETWEEN_RETRIES) timeToWait =
TIME_BETWEEN_RETRIES;// we dont want to wait for more than
TIME_BETWEEN_RETRIES second each time
```

[Table of Contents] [Description]

Path 511:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
305. public static void closeConnection(Connection connection) //dbutils.java
306. if (connection == null) return;
```

[Table of Contents] [Description]

Path 512:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
349. public static void closeStatement(Statement statement) //dbutils.java
     ...
351. if (statement != null) statement.close();
```

[Table of Contents] [Description]

Path 513:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
349. public static void closeStatement(Statement statement) //dbutils.java
     ...
351. if (statement != null) statement.close();
```

[Table of Contents] [Description]

Path 514:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
361. public static void closeResultSet(ResultSet rs) //dbutils.java
     ...
363. if (rs != null) rs.close();
```

[Table of Contents] [Description]

Path 515:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
361. public static void closeResultSet(ResultSet rs) //dbutils.java
       ...
363. if (rs != null) rs.close();
```

[Table of Contents] [Description]

Path 516:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
229. public static void closeConnection(Connection connection) //dbutils2.java
       230. if (connection == null) return;
```

[Table of Contents] [Description]

Path 517:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //disablehtmltagfilter.java
       ...
63. } else if ( ((i + 3) < length) &&
```

[Table of Contents] [Description]

Path 518:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //disablehtmltagfilter.java
       ...
69. } else if ( ((i + 3) < length) &&
```

[Table of Contents] [Description]

Path 519:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //disablehtmltagfilter.java
       ...
75. } else if ( ((i + 4) < length) &&
```

[Table of Contents] [Description]

Path 520:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //disablehtmltagfilter.java
      ...
82. } else if ( ((i + 5) < length) &&
```

[Table of Contents] [Description]

Path 521:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
142. public static String filter(String input, String
emotionFolder) //enableemotionfilter.java
      143. if (input == null) return null;
```

[Table of Contents] [Description]

Path 522:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
142. public static String filter(String input, String
emotionFolder) //enableemotionfilter.java
      ...
172. if (endIndex > inputLength) continue;
```

[Table of Contents] [Description]

Path 523:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //enablehtmltagfilter.java
      ...
52. if ( ((i + 3) < length) &&
```

[Table of Contents] [Description]

Path 524:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //enablehtmltagfilter.java
      ...
58. } else if ( ((i + 3) < length) &&
```

[Table of Contents] [Description]

Path 525:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //enablehtmltagfilter.java
      ...
64. } else if ( ((i + 4) < length) &&
```

[Table of Contents] [Description]

Path 526:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
41. public static String filter(String input) //enablehtmltagfilter.java
      ...
71. } else if ( ((i + 5) < length) &&
```

[Table of Contents] [Description]

Path 527:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
163. public static String filter(String input) //enablemvncodefilter.java
      ...
187. if (endIndex > inputLength) continue;
```

[Table of Contents] [Description]

Path 528:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
55. if (input == null || input.length() == 0)
```

[Table of Contents] [Description]

Path 529:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
67. if ((cur == 'f' && index < len - 6 && chars[++i] == 't' && chars[++i] ==
'p' ||
```

[Table of Contents] [Description]

Path 530:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
75. if (cur == ' ' || cur == '\t' || cur == '\'' ||
```

[Table of Contents] [Description]

Path 531:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
84. if (cur == '.' || cur == ',' || cur == ')' || cur == ']')
```

[Table of Contents] [Description]

Path 532:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
84. if (cur == '.' || cur == ',' || cur == ')' || cur == ']')
```

[Table of Contents] [Description]

Path 533:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
174. if (oldend < len)
```

[Table of Contents] [Description]

Path 534:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
54. public static String filter(String input) //urlfilter.java
      ...
174. if (oldend < len)
```

[Table of Contents] [Description]

Path 535:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
63. public static boolean filter(HttpServletRequest request) //useragentfilter.java
      ...
70. if (checkUserAgent == null) return true;
```

[Table of Contents] [Description]

Path 536:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
168. public static boolean isArrayByteBase64( byte[] arrayOctect ) //base64.java
       ...
179. if ( !Base64.isBase64(arrayOctect[i]) )
```

[Table of Contents] [Description]

Path 537:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
70. public static synchronized String getMD5_Base64(String input) //encoder.java
      ...
82. if (digest == null)
```

[Table of Contents] [Description]

Path 538:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
254. public void write(int b) throws IOException //compressionresponsestream.java
       ...
257. if (closed)
```

[Table of Contents] [Description]

Path 539:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
294. public void write(byte b[], int off, int len) throws
IOException //compressionresponsestream.java
       ...
304. if (closed)
```

[Table of Contents] [Description]

Path 540:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
294. public void write(byte b[], int off, int len) throws
IOException //compressionresponsestream.java
      ...
307. if (len == 0)
```

[Table of Contents] [Description]

Path 541:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
      ...
200. if (stream != null)
```

[Table of Contents] [Description]

Path 542:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
      ...
200. if (stream != null)
```

[Table of Contents] [Description]

Path 543:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
235. public ServletOutputStream getOutputStream() throws
IOException //compressionservletresponsewrapper.java
      ...
237. if (writer != null)
```

[Table of Contents] [Description]

Path 544:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
256. public PrintWriter getWriter() throws
IOException //compressionservletresponsewrapper.java
      ...
258. if (writer != null)
```

[Table of Contents] [Description]

Path 545:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
256. public PrintWriter getWriter() throws
IOException //compressionservletresponsewrapper.java
     ...
261. if (stream != null)
```

[Table of Contents] [Description]

Path 546:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
180. public void init(FilterConfig filterConfig) throws
ServletException //setcharacterencodingfilter.java
     ...
193. if (value == null)
```

[Table of Contents] [Description]

Path 547:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
180. public void init(FilterConfig filterConfig) throws
ServletException //setcharacterencodingfilter.java
     ...
197. else if (value.equals(""))
```

[Table of Contents] [Description]

Path 548:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
180. public void init(FilterConfig filterConfig) throws
ServletException //setcharacterencodingfilter.java
     ...
201. else if (value.equalsIgnoreCase("true"))
```

[Table of Contents] [Description]

Path 549:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
180. public void init(FilterConfig filterConfig) throws
ServletException //setcharacterencodingfilter.java
```

```
...
203. else if (value.equalsIgnoreCase("yes"))
```

[Table of Contents] [Description]

Path 550:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
404. if (Quality <= 0)
```

[Table of Contents] [Description]

Path 551:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
406. if (Quality > 100)
```

[Table of Contents] [Description]

Path 552:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
408. if (Quality < 50)
```

[Table of Contents] [Description]

Path 553:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
483. if ( temp <= 0) temp = 1;
```

[Table of Contents] [Description]

Path 554:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
```

```
...
484. if (temp > 255) temp = 255;
```

[Table of Contents] [Description]

Path 555:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
571. if ( temp <= 0) temp = 1;
```

[Table of Contents] [Description]

Path 556:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
390. private void initMatrix(int quality) //jpegencoder.java
...
572. if (temp >= 255) temp = 255;
```

[Table of Contents] [Description]

Path 557:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
93. public void write(String value) throws IOException //csvutil.java
...
129. if (autoFlush) flush();
```

[Table of Contents] [Description]

Path 558:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
93. public void write(String value) throws IOException //csvutil.java
...
129. if (autoFlush) flush();
```

[Table of Contents] [Description]

Path 559:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
146. public void writeln() throws IOException //csvutil.java
```

```
...
149. if (autoFlush) flush();
```

[Table of Contents] [Description]

Path 560:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
146. public void writeln() throws IOException //csvutil.java
...
149. if (autoFlush) flush();
```

[Table of Contents] [Description]

Path 561:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
192. public void writelnComment(String comment) throws IOException //csvutil.java
...
194. if (comment==null) comment = "";
```

[Table of Contents] [Description]

Path 562:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
136. public static void deleteDir(File dir) throws IOException //fileutil.java
137. if (dir.isFile()) throw new IOException("IOException ->
BadInputException: not a directory.");
```

[Table of Contents] [Description]

Path 563:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
152. public static long getDirLength(File dir) throws IOException //fileutil.java
153. if (dir.isFile()) throw new IOException("BadInputException: not a
directory.");
```

[Table of Contents] [Description]

Path 564:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
171. public static long getDirLength_onDisk(File dir) throws
```

```
IOException //fileutil.java
      172. if (dir.isFile()) throw new IOException("BadInputException: not a
      directory.");
```

[Table of Contents] [Description]

Path 565:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
171. public static long getDirLength_onDisk(File dir) throws
IOException //fileutil.java
      ...
      185. if (mod == 0) mod = 1;
```

[Table of Contents] [Description]

Path 566:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
      ...
      218. if (destFile.exists()) return;
```

[Table of Contents] [Description]

Path 567:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
      ...
      239. if (readLength == -1) break;// end of file
```

[Table of Contents] [Description]

Path 568:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
261. public static byte[] getBytes(InputStream inputStream) throws
IOException //fileutil.java
      ...
      267. if (readLength == -1) break;// end of file
```

[Table of Contents] [Description]

Path 569:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
298. public static void popFile(File srcFile, OutputStream output) throws
IOException //fileutil.java
     |...
     |306. if (length == -1) break;// end of file
```

[Table of Contents] [Description]

Path 570:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
381. public static void createTextFile(InputStream inputStream, String
textFile) //fileutil.java
     |...
     |398. if (outputStream != null) outputStream.close();
```

[Table of Contents] [Description]

Path 571:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
381. public static void createTextFile(InputStream inputStream, String
textFile) //fileutil.java
     |...
     |398. if (outputStream != null) outputStream.close();
```

[Table of Contents] [Description]

Path 572:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
     |...
     |442. if (out != null) out.close();
```

[Table of Contents] [Description]

Path 573:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
     |...
```

```
442. if (out != null) out.close();
```

[Table of Contents] [Description]

Path 574:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
      ...
469. if (readLength == -1) break;// end of file
```

[Table of Contents] [Description]

Path 575:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
      ...
481. if (reader != null) reader.close();
```

[Table of Contents] [Description]

Path 576:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
      ...
481. if (reader != null) reader.close();
```

[Table of Contents] [Description]

Path 577:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
linesToReturn) //fileutil.java
      ...
535. if (start < 0) start = 0; // not start line
```

[Table of Contents] [Description]

Path 578:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      |...
      |60. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 579:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      |...
      |72. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 580:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
196. public static boolean getParameterBoolean(GenericRequest request, String
param) //genericparamutil.java
      |...
      |199. if (inputStr.length() == 0) return false;
```

[Table of Contents] [Description]

Path 581:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public static String getParameterPassword(GenericRequest request, String param,
int minLength, int option) //genericparamutil.java
      |...
      |252. if (minLength < 1) minLength = 1;
```

[Table of Contents] [Description]

Path 582:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
249. public static String getParameterPassword(GenericRequest request, String param,
int minLength, int option) //genericparamutil.java
      |...
      |255. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 583:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
379. public static String getAttribute(HttpSession session, String
name) //genericparamutil.java
      |
      ...
      382. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 584:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
394. public static String getAttribute(GenericRequest request, String
name) //genericparamutil.java
      |
      ...
      397. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 585:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
188. public static Image getFitSizeImage(Image srcImage, int fitWidth, int
fitHeight) //imageutil.java
      |
      ...
      190. if ((fitWidth < 100) || (fitHeight < 100)) throw new
      IllegalArgumentException("Cannot accept values < 100");
```

[Table of Contents] [Description]

Path 586:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
188. public static Image getFitSizeImage(Image srcImage, int fitWidth, int
fitHeight) //imageutil.java
      |
      ...
      197. if ((srcWidth == fitWidth) && (srcHeight == fitHeight)) return srcImage;
```

[Table of Contents] [Description]

Path 587:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
418. public void setDimensions (int w, int h) //imageutil.java
       ...
432. else if (destHeight < 0)
```

[Table of Contents] [Description]

Path 588:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
445. public void imageComplete (int status) //imageutil.java
       ...
447. if ( status == STATICIMAGEDONE
```

[Table of Contents] [Description]

Path 589:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
445. public void imageComplete (int status) //imageutil.java
       ...
447. if ( status == STATICIMAGEDONE
```

[Table of Contents] [Description]

Path 590:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
453. public void setPixels (int x, int y, int width, int height, //imageutil.java
       ...
457. if (srcPixels == null)
```

[Table of Contents] [Description]

Path 591:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
470. public void setPixels (int x, int y, int width, int height, //imageutil.java
       ...
474. if (srcPixels == null)
```

[Table of Contents] [Description]

Path 592:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
489. private int [] calcRow () //imageutil.java
       ...
492. if (destPixels == null)
```

[Table of Contents] [Description]

Path 593:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
511. private void accumPixels (int x, int y, int w, int h, //imageutil.java
       ...
565. if ((sxrem -= amtx) == 0)
```

[Table of Contents] [Description]

Path 594:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
608. public static int roundDiv(int dividend, int divisor) //imageutil.java
       ...
610. if (Math.abs(remainder) * 2 <= Math.abs(divisor))
```

[Table of Contents] [Description]

Path 595:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
608. public static int roundDiv(int dividend, int divisor) //imageutil.java
       ...
613. if (dividend * divisor < 0)
```

[Table of Contents] [Description]

Path 596:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
626. public static long roundDiv (long dividend, long divisor) //imageutil.java
       ...
628. if (Math.abs (remainder) * 2 <= Math.abs (divisor))
```

[Table of Contents] [Description]

Path 597:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
626. public static long roundDiv (long dividend, long divisor) //imageutil.java
       ...
631. if (dividend * divisor < 0)
```

[Table of Contents] [Description]

Path 598:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
73. public static String getEmailUsername(String email) //mailutil.java
      74. if (email == null) return "";
```

[Table of Contents] [Description]

Path 599:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
87. public static String getEmailDomain(String email) //mailutil.java
      88. if (email == null) return "";
```

[Table of Contents] [Description]

Path 600:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
102. public static void checkGoodEmail(String input) throws
BadInputException //mailutil.java
       103. if (input == null) throw new BadInputException("Sorry, null string is not
       a good email.");//@todo : localize me
```

[Table of Contents] [Description]

Path 601:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
290. if (from == null) from = MVNCoreConfig.getDefaultMailFrom();
```

[Table of Contents] [Description]

Path 602:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
```

```
...
354. if (transport != null) transport.close();
```

[Table of Contents] [Description]

Path 603:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
...
354. if (transport != null) transport.close();
```

[Table of Contents] [Description]

Path 604:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
...
391. if (transport != null) transport.close();
```

[Table of Contents] [Description]

Path 605:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
...
391. if (transport != null) transport.close();
```

[Table of Contents] [Description]

Path 606:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
413. public static String[] getEmails(String email) throws
BadInputException //mailutil.java
414. if (email == null) email = "";
```

[Table of Contents] [Description]

Path 607:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
459. private static InternetAddress[] getInternetAddressEmails(String
```

```
email) //mailutil.java
    ...
462. if (mails.length == 0) return null;// must return null, not empty array
```

[Table of Contents] [Description]

Path 608:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
100. public static String getParameter(HttpServletRequest request, String
param) //paramutil.java
    ...
103. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 609:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
115. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 610:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
239. public static boolean getParameterBoolean(HttpServletRequest request, String
param) //paramutil.java
    ...
242. if (inputStr.length() == 0) return false;
```

[Table of Contents] [Description]

Path 611:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
292. public static String getParameterPassword(HttpServletRequest request, String
param, int minLength, int option) //paramutil.java
    ...
295. if (minLength < 1) minLength = 1;
```

[Table of Contents] [Description]

Path 612:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
292. public static String getParameterPassword(HttpServletRequest request, String
param, int minLength, int option) //paramutil.java
      ...
297. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 613:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
392. public static String getAttribute(HttpSession session, String
name) //paramutil.java
      ...
395. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 614:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
407. public static String getAttribute(ServletRequest request, String
name) //paramutil.java
      ...
410. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 615:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
423. public static String getAttribute(HttpServletRequest request, String
name) //paramutil.java
      ...
426. if (ret == null) ret = "";
```

[Table of Contents] [Description]

Path 616:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
66. public static String[] getStringArray(String inputValue, String
delim) //stringutil.java
      67. if (inputValue == null) inputValue = "";
```

[Table of Contents] [Description]

Path 617:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     |122. if (str == null) return "";
```

[Table of Contents] [Description]

Path 618:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
183. public static String getShorterString(String str, int
maxLength) //stringutil.java
     |...
     |185. if (maxLength < 0) throw new IllegalArgumentException("The maxLength < 0
     is not allowed.");
```

[Table of Contents] [Description]

Path 619:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
212. public static String getShorterStringIgnoreSpace(String str, int
maxLength) //stringutil.java
     |213. if (maxLength < 0) throw new IllegalArgumentException("The maxLength < 0
     is not allowed.");
```

[Table of Contents] [Description]

Path 620:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
212. public static String getShorterStringIgnoreSpace(String str, int
maxLength) //stringutil.java
     |...
     |214. if (str == null) return "";
```

[Table of Contents] [Description]

Path 621:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
212. public static String getShorterStringIgnoreSpace(String str, int
maxLength) //stringutil.java
```

```
...
215. if (str.length() <= maxLength) return str;
```

[Table of Contents] [Description]

Path 622:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
272. public static String getHiddenPassword(String password) //stringutil.java
...
275. if (length == 0) return password;
```

[Table of Contents] [Description]

Path 623:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
300. public static String escapeQuote(String input) //stringutil.java
...
312. if ( i + 5 < chars.length && // find '&apos;' ~ single quote
```

[Table of Contents] [Description]

Path 624:

Query Name - **Single_Line_If_Statement**

Severity - **Information**

```
300. public static String escapeQuote(String input) //stringutil.java
...
322. } else if ( i + 4 < chars.length && // find '&#39;' ~ single quote
```

[Table of Contents] [Description]

Improper Error Handling

CWE ID	390
Description	Sometimes an error is detected, and bad or no action is taken.
Likelihood of Exploit	Medium
Potential Mitigations	Implementation: Properly handle each exception. This is the recommended solution. Ensure that all exceptions are handled in such a way that you can be sure of the state of your system at any given moment. Subject the software to extensive testing to discover some of the possible instances of where/how errors or return values are not handled. Consider testing techniques such as ad hoc, equivalence partitioning, robustness and fault tolerance, mutation, and fuzzing.

Demonstrative Examples

C Example:

```
foo=malloc(sizeof(char); //the next line checks to see if malloc failed
if (foo==0) {
  //We do nothing so we just ignore the error.
}
```

C++ Example:

```
while (DoSomething()) {
  try {
    /* perform main loop here */
  }
  catch (Exception e) {
    /* do nothing, but catch so it'll compile... */
  }
}
```

Java Example:

```
while (DoSomething()) {
  try {
    /* perform main loop here */
  }
  catch (Exception e) {
    /* do nothing, but catch so it'll compile... */
  }
}
```

Context Notes If a function returns an error, it is important to either fix the problem and try again, alert the user that an error has happened and let the program continue, or alert the user and close and cleanup the program.

Node Relationships ChildOf - Error Conditions, Return Values, Status Codes (389)

Source Taxonomies CLASP - Improper error handling

Applicable Platforms All

Path 1:

Query Name - **Empty_Catch**

Severity - **Information**

```
198. public final static void waitln() //utility.java
       ...
202. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 2:

Query Name - **Empty_Catch**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
      |
      ...
      |
582. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 3:

Query Name - **Empty_Catch**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
      |
      ...
      |
590. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 4:

Query Name - **Empty_Catch**

Severity - **Information**

```
537. public static void migrateThreads(PrintStream ostream) //migrator.java
      |
      ...
      |
603. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **Empty_Catch**

Severity - **Information**

```
659. public static void migratePrivateMessages(PrintStream ostream) //migrator.java
      |
      ...
      |
696. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 6:

Query Name - **Empty_Catch**

Severity - **Information**

```
659. public static void migratePrivateMessages(PrintStream ostream) //migrator.java
      |
      ...
      |
702. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 7:

Query Name - **Empty_Catch**

Severity - **Information**

```
120. public void create(int categoryID, int parentCategoryID, String
categoryName, //categorydaoimpljdbc.java
     ...
133. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 8:

Query Name - **Empty_Catch**

Severity - **Information**

```
120. public void create(int categoryID, int parentCategoryID, String
categoryName, //categorydaoimpljdbc.java
     ...
144. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 9:

Query Name - **Empty_Catch**

Severity - **Information**

```
123. public void create(int forumID, int categoryID, String
lastPostMemberName, //forumdaoimpljdbc.java
     ...
139. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 10:

Query Name - **Empty_Catch**

Severity - **Information**

```
123. public void create(int forumID, int categoryID, String
lastPostMemberName, //forumdaoimpljdbc.java
     ...
150. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 11:

Query Name - **Empty_Catch**

Severity - **Information**

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
     ...
98. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 12:

Query Name - **Empty_Catch**

Severity - **Information**

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
      ...
133. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 13:

Query Name - **Empty_Catch**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
      ...
204. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 14:

Query Name - **Empty_Catch**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
      ...
215. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 15:

Query Name - **Empty_Catch**

Severity - **Information**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
      ...
226. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 16:

Query Name - **Empty_Catch**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpl.java
      ...
```

```
442. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 17:

Query Name - **Empty_Catch**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
...
202. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 18:

Query Name - **Empty_Catch**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
...
213. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 19:

Query Name - **Empty_Catch**

Severity - **Information**

```
175. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpljdbc.java
...
224. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 20:

Query Name - **Empty_Catch**

Severity - **Information**

```
415. public void update(int memberID, // primary key //memberdaoimpljdbc.java
...
442. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 21:

Query Name - **Empty_Catch**

Severity - **Information**

```
630. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int
memberEmailVisible, //memberdaoimpljdbc.java
      ...
648. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 22:

Query Name - **Empty_Catch**

Severity - **Information**

```
630. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int
memberEmailVisible, //memberdaoimpljdbc.java
      ...
659. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 23:

Query Name - **Empty_Catch**

Severity - **Information**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
95. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 24:

Query Name - **Empty_Catch**

Severity - **Information**

```
93. public void create(String folderName, int memberID, int folderOrder, int
folderStatus, int folderOption, //messagefolderdaoimpljdbc.java
      ...
101. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 25:

Query Name - **Empty_Catch**

Severity - **Information**

```
94. public void create(int postID, int parentPostID, int
forumID, //postdaoimpljdbc.java
      ...
111. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 26:

Query Name - **Empty_Catch**

Severity - **Information**

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
     ...
133. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 27:

Query Name - **Empty_Catch**

Severity - **Information**

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
     ...
144. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 28:

Query Name - **Empty_Catch**

Severity - **Information**

```
93. public int createThread(int threadID, int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
     ...
109. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 29:

Query Name - **Empty_Catch**

Severity - **Information**

```
122. public void create(int memberID, int categoryID, int forumID, int threadID, int
watchType, int watchOption, //watchdaoimpljdbc.java
     ...
150. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 30:

Query Name - **Empty_Catch**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
      ...
331. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 31:

Query Name - **Empty_Catch**

Severity - **Information**

```
213. public void exportCategoryList() //importexportxml.java
      ...
344. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 32:

Query Name - **Empty_Catch**

Severity - **Information**

```
93. public void contextDestroyed(ServletContextEvent
event) //mvnforumcontextlistener.java
      ...
104. } catch (InterruptedException e) {
```

[Table of Contents] [Description]

Path 33:

Query Name - **Empty_Catch**

Severity - **Information**

```
423. public static void writeMvnForumImage(HttpServletResponse response) throws
IOException //myutil.java
      ...
441. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 34:

Query Name - **Empty_Catch**

Severity - **Information**

```
446. public static void writeMvnCoreImage(HttpServletResponse response) throws
IOException //myutil.java
      ...
465. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 35:

Query Name - **Empty_Catch**

Severity - **Information**

```
314. public static void exportAttachmentList(XMLWriter xmlWriter, int
parentPostID) //attachmentxml.java
     ...
369. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 36:

Query Name - **Empty_Catch**

Severity - **Information**

```
239. public static void exportCategoryWatchesForCategory(XMLWriter xmlWriter, int
categoryID) //categoryxml.java
     ...
282. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 37:

Query Name - **Empty_Catch**

Severity - **Information**

```
341. public static void exportSubCategoryList(XMLWriter xmlWriter, int
parentCategoryID) //categoryxml.java
     ...
364. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 38:

Query Name - **Empty_Catch**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
     ...
175. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 39:

Query Name - **Empty_Catch**

Severity - **Information**

```
136. public static void getExportXmlZip(HttpServletRequest request,
HttpServletResponse response) //exportwebhandler.java
     ...
183. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 40:

Query Name - **Empty_Catch**

Severity - **Information**

```
241. private static void sendToUser(BufferedInputStream
inputStream, //exportwebhandler.java
     ...
254. } catch (IOException ee) {
```

[Table of Contents] [Description]

Path 41:

Query Name - **Empty_Catch**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
     ...
182. try { out.close(); } catch (Exception ee) {}
```

[Table of Contents] [Description]

Path 42:

Query Name - **Empty_Catch**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
     ...
184. try { new File(zipFilePath).delete(); } catch (Exception ee) {}
```

[Table of Contents] [Description]

Path 43:

Query Name - **Empty_Catch**

Severity - **Information**

```
103. public void process(HttpServletRequest request, HttpServletResponse
response) //forumadminservlet.java
     ...
169. } catch (AuthenticationException e) {
```

[Table of Contents] [Description]

Path 44:

Query Name - **Empty_Catch**

Severity - **Information**

```
218. public void addMemberForumPermission(String memberName, String
permission) //forumxml.java
     ...
238. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **Empty_Catch**

Severity - **Information**

```
259. public void addGroupForumPermission(String groupName, String
permission) //forumxml.java
     ...
279. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 46:

Query Name - **Empty_Catch**

Severity - **Information**

```
284. public void addGuestMemberForumPermission(String permission) //forumxml.java
     ...
299. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 47:

Query Name - **Empty_Catch**

Severity - **Information**

```
304. public void addRegisteredMembersGroupForumPermission(String
permission) //forumxml.java
     ...
319. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 48:

Query Name - **Empty_Catch**

Severity - **Information**

```
425. public static void exportForumWatchesForForum(XMLWriter xmlWriter, int
forumID) //forumxml.java
     ...
467. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 49:

Query Name - **Empty_Catch**

Severity - **Information**

```
474. public static void exportMemberForumPermissionsForForum(XMLWriter xmlWriter,
int forumID) //forumxml.java
     ...
500. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **Empty_Catch**

Severity - **Information**

```
507. public static void exportGroupForumPermissionsForForum(XMLWriter xmlWriter, int
forumID) //forumxml.java
     ...
532. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 51:

Query Name - **Empty_Catch**

Severity - **Information**

```
625. public static void exportForumList(XMLWriter xmlWriter, int
parentCategoryID) //forumxml.java
     ...
647. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 52:

Query Name - **Empty_Catch**

Severity - **Information**

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
     ...
801. } catch (IOException e) { }
```

[Table of Contents] [Description]

Path 53:

Query Name - **Empty_Catch**

Severity - **Information**

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
```

```
...
95. } catch (ForeignKeyNotFoundException ex) {
```

[Table of Contents] [Description]

Path 54:

Query Name - **Empty_Catch**

Severity - **Information**

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
...
98. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 55:

Query Name - **Empty_Catch**

Severity - **Information**

```
189. public void processUpdateGroupOwner(GenericRequest
request) //groupswebhandler.java
...
231. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 56:

Query Name - **Empty_Catch**

Severity - **Information**

```
207. public void addGroupPermission(String permission) //groupxml.java
...
220. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 57:

Query Name - **Empty_Catch**

Severity - **Information**

```
295. public static void exportGlobalPermissionsForGroup(XMLWriter xmlWriter, int
groupID) //groupxml.java
...
314. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 58:

Query Name - **Empty_Catch**

Severity - **Information**

```
321. public static void exportGroupMembersForGroup(XMLWriter xmlWriter, int
groupID) //groupxml.java
     ...
352. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 59:

Query Name - **Empty_Catch**

Severity - **Information**

```
410. public static void exportGroupList(XMLWriter xmlWriter) //groupxml.java
     ...
422. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 60:

Query Name - **Empty_Catch**

Severity - **Information**

```
410. public static void exportGroupList(XMLWriter xmlWriter) //groupxml.java
     ...
439. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 61:

Query Name - **Empty_Catch**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
     ...
283. } catch (Exception e) {}
```

[Table of Contents] [Description]

Path 62:

Query Name - **Empty_Catch**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
     ...
334. } catch (ImportException e) {
```

[Table of Contents] [Description]

Path 63:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public void processAdd(GenericRequest request) //membergroupwebhandler.java
      ...
118. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 64:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public void processAdd(GenericRequest request) //membergroupwebhandler.java
      ...
120. } catch (ForeignKeyNotFoundException ex) {
```

[Table of Contents] [Description]

Path 65:

Query Name - **Empty_Catch**

Severity - **Information**

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
      ...
651. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 66:

Query Name - **Empty_Catch**

Severity - **Information**

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
      ...
720. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 67:

Query Name - **Empty_Catch**

Severity - **Information**

```
738. public void prepareListPendingUsers(GenericRequest
request) //memberwebhandler.java
      ...
758. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 68:

Query Name - **Empty_Catch**

Severity - **Information**

```
462. public void addMemberPermission(String permission) //memberxml.java
       ...
475. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 69:

Query Name - **Empty_Catch**

Severity - **Information**

```
480. public static void addGuestMemberPermission(String permission) //memberxml.java
       ...
490. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 70:

Query Name - **Empty_Catch**

Severity - **Information**

```
495. public static void addAdminMemberPermission(String permission) //memberxml.java
       ...
505. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 71:

Query Name - **Empty_Catch**

Severity - **Information**

```
510. public static void addMemberPermission(String memberName, String
permission) //memberxml.java
       ...
521. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 72:

Query Name - **Empty_Catch**

Severity - **Information**

```
640. public static void exportMessageFoldersForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
```

```
...
670. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 73:

Query Name - **Empty_Catch**

Severity - **Information**

```
687. public static void exportGlobalPermissionsForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
...
706. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 74:

Query Name - **Empty_Catch**

Severity - **Information**

```
723. public static void exportGlobalWatchesForMember(XMLWriter xmlWriter, int
memberID) //memberxml.java
...
760. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 75:

Query Name - **Empty_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
...
993. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 76:

Query Name - **Empty_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
...
995. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 77:

Query Name - **Empty_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
       ...
998. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 78:

Query Name - **Empty_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
       ...
1000. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 79:

Query Name - **Empty_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
       ...
1016. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 80:

Query Name - **Empty_Catch**

Severity - **Information**

```
351. public static void exportPostList_FirstPosts(XMLWriter xmlWriter, int
parentThreadID) //postxml.java
       ...
374. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 81:

Query Name - **Empty_Catch**

Severity - **Information**

```
381. public static void exportPostList_Replies(XMLWriter xmlWriter, int
parentPostID) //postxml.java
       ...
403. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 82:

Query Name - **Empty_Catch**

Severity - **Information**

```
133. public static void exportRankList(XMLWriter xmlWriter) //rankxml.java
     ...
169. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 83:

Query Name - **Empty_Catch**

Severity - **Information**

```
339. public static void exportThreadWatchesForThread(XMLWriter xmlWriter, int
threadID) //threadxml.java
     ...
380. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 84:

Query Name - **Empty_Catch**

Severity - **Information**

```
387. public static void exportFavoriteThreadsForThread(XMLWriter xmlWriter, int
threadID) //threadxml.java
     ...
423. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 85:

Query Name - **Empty_Catch**

Severity - **Information**

```
523. public static void exportThreadList(XMLWriter xmlWriter, int
parentForumID) //threadxml.java
     ...
545. } catch (NoSuchElementException e) {
```

[Table of Contents] [Description]

Path 86:

Query Name - **Empty_Catch**

Severity - **Information**

```
184. protected static void processXml(File inputFile, Calendar
importTime) //importjive.java
     ...
376. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 87:

Query Name - **Empty_Catch**

Severity - **Information**

```
184. protected static void processXml(File inputFile, Calendar
importTime) //importjive.java
      ...
388. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 88:

Query Name - **Empty_Catch**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      ...
173. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 89:

Query Name - **Empty_Catch**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      ...
188. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 90:

Query Name - **Empty_Catch**

Severity - **Information**

```
158. public void addJiveForumUser(String usertype, String username, String
jivePermission) //jiveforumxml.java
      ...
203. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 91:

Query Name - **Empty_Catch**

Severity - **Information**

```
216. public void addJiveForumGroup(String groupname, String
```

```
jivePermission) //jiveforumxml.java
    ...
230. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 92:

Query Name - **Empty_Catch**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
    ...
226. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 93:

Query Name - **Empty_Catch**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
    ...
240. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 94:

Query Name - **Empty_Catch**

Severity - **Information**

```
215. public void addJiveUserPermission(String usertype, String username, String
jivePermission) //jivexml.java
    ...
254. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 95:

Query Name - **Empty_Catch**

Severity - **Information**

```
268. public void addJiveGroupPermission(String groupname, String
jivePermission) //jivexml.java
    ...
278. } catch (DuplicateKeyException e) {
```

[Table of Contents] [Description]

Path 96:

Query Name - **Empty_Catch**

Severity - **Information**

```
188. public static void checkGuestMember() throws CreateException,
DuplicateKeyException, //mvnforumxml.java
      ...
198. } catch (DuplicateKeyException de) {
```

[Table of Contents] [Description]

Path 97:

Query Name - **Empty_Catch**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
      ...
275. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 98:

Query Name - **Empty_Catch**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
      ...
282. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 99:

Query Name - **Empty_Catch**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
      ...
289. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 100:

Query Name - **Empty_Catch**

Severity - **Information**

```
263. public void postLogin(HttpServletRequest request, HttpServletResponse response,
OnlineUser onlineUser) //onlineuserfactoryimpl.java
```

```
...
296. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 101:

Query Name - **Empty_Catch**

Severity - **Information**

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
...
309. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 102:

Query Name - **Empty_Catch**

Severity - **Information**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
437. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 103:

Query Name - **Empty_Catch**

Severity - **Information**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
457. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 104:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
523. } catch (AuthenticationException e) {
```

[Table of Contents] [Description]

Path 105:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
550. } catch (AuthenticationException e) {
```

[Table of Contents] [Description]

Path 106:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
564. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 107:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
581. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 108:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
602. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 109:

Query Name - **Empty_Catch**

Severity - **Information**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
```

```
622. } catch (ObjectNotFoundException oe) {
```

[Table of Contents] [Description]

Path 110:

Query Name - **Empty_Catch**

Severity - **Information**

```
258. public String drawCategory(CategoryTreeEvent event) //categoryselector.java
     ...
268. } catch (Exception ex) {}
```

[Table of Contents] [Description]

Path 111:

Query Name - **Empty_Catch**

Severity - **Information**

```
288. public String drawCategory(CategoryTreeEvent event) //listforums.java
     ...
314. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 112:

Query Name - **Empty_Catch**

Severity - **Information**

```
172. public void writeCaptchaImage(HttpServletRequest request, HttpServletResponse
response) //mvncaptchaservice.java
     ...
203. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 113:

Query Name - **Empty_Catch**

Severity - **Information**

```
788. public Collection getAttachments_withSortSupport_limit_general(int offset, int
rowsToReturn, int category, int forum) //attachmentdaoimpljdbc.java
     ...
826. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 114:

Query Name - **Empty_Catch**

Severity - **Information**

```
144. public void create(int parentCategoryID, String categoryName, String
categoryDesc, //categorydaoimpljdbc.java
     ...
156. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 115:

Query Name - **Empty_Catch**

Severity - **Information**

```
236. public void update(int categoryID, // primary key //categorydaoimpljdbc.java
     ...
248. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 116:

Query Name - **Empty_Catch**

Severity - **Information**

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
     ...
120. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 117:

Query Name - **Empty_Catch**

Severity - **Information**

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
     ...
164. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 118:

Query Name - **Empty_Catch**

Severity - **Information**

```
289. public void update(int forumID, // primary key //forumdaoimpljdbc.java
     ...
311. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 119:

Query Name - **Empty_Catch**

Severity - **Information**

```
918. public Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      952. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 120:

Query Name - **Empty_Catch**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1033. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 121:

Query Name - **Empty_Catch**

Severity - **Information**

```
1079. public Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1113. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 122:

Query Name - **Empty_Catch**

Severity - **Information**

```
1180. public Collection getBeans_withSortSupport_limit_ViewCount_mysql(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
      ...
      1219. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 123:

Query Name - **Empty_Catch**

Severity - **Information**

```
1262. public Collection getBeans_withSortSupport_limit_ViewCount_oracle(int offset,
int rowsToReturn, String sort, String order) throws IllegalArgumentException,
```

```
DatabaseException //forumdaoimpljdbc.java
    ...
1301. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 124:

Query Name - **Empty_Catch**

Severity - **Information**

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
    ...
116. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 125:

Query Name - **Empty_Catch**

Severity - **Information**

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
    ...
115. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 126:

Query Name - **Empty_Catch**

Severity - **Information**

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
    ...
157. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 127:

Query Name - **Empty_Catch**

Severity - **Information**

```
242. public void update(int groupID, // primary key //groupsdaoimpljdbc.java
    ...
253. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 128:

Query Name - **Empty_Catch**

Severity - **Information**

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
      ...
285. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 129:

Query Name - **Empty_Catch**

Severity - **Information**

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
      ...
296. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 130:

Query Name - **Empty_Catch**

Severity - **Information**

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
      ...
491. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 131:

Query Name - **Empty_Catch**

Severity - **Information**

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
1815. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 132:

Query Name - **Empty_Catch**

Severity - **Information**

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
      ...
2176. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 133:

Query Name - **Empty_Catch**

Severity - **Information**

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      |...
      |116. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 134:

Query Name - **Empty_Catch**

Severity - **Information**

```
107. public void create(int groupID, String
memberName, //membergroupdaoimpljdbc.java
      |...
      |129. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 135:

Query Name - **Empty_Catch**

Severity - **Information**

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      |...
      |115. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 136:

Query Name - **Empty_Catch**

Severity - **Information**

```
436. public Collection getBeans_inMember_inFolder_withSortSupport_limit_general(int
memberID, String folderName, int offset, int rowsToReturn, String sort, String
order, boolean onlyNonPublic) //messagedaoimpljdbc.java
      |...
      |463. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 137:

Query Name - **Empty_Catch**

Severity - **Information**

```
107. public void create(String folderName, int memberID, int
folderOrder, //messagefolderdaoimpljdbc.java
     ...
120. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 138:

Query Name - **Empty_Catch**

Severity - **Information**

```
106. public void create(int messageID, int pmAttachID, int
relationType, //pmattachmessagedaoimpljdbc.java
     ...
118. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 139:

Query Name - **Empty_Catch**

Severity - **Information**

```
822. private Collection getBeans_inThread_limit_general(int threadID, int offset,
int rowsToReturn, boolean enable) //postdaoimpljdbc.java
     ...
847. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 140:

Query Name - **Empty_Catch**

Severity - **Information**

```
1078. public Collection getLastEnablePosts_inThread_limit(int threadID, int
rowsToReturn) //postdaoimpljdbc.java
     ...
1097. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 141:

Query Name - **Empty_Catch**

Severity - **Information**

```
1146. public Collection getLastEnablePosts_inForum_limit(int forumID, int
rowsToReturn) //postdaoimpljdbc.java
     ...
1165. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 142:

Query Name - **Empty_Catch**

Severity - **Information**

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
      ...
148. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 143:

Query Name - **Empty_Catch**

Severity - **Information**

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
      ...
159. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 144:

Query Name - **Empty_Catch**

Severity - **Information**

```
197. public void update(int rankID, // primary key //rankdaoimpljdbc.java
      ...
209. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 145:

Query Name - **Empty_Catch**

Severity - **Information**

```
197. public void update(int rankID, // primary key //rankdaoimpljdbc.java
      ...
219. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 146:

Query Name - **Empty_Catch**

Severity - **Information**

```
1194. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, boolean enable, boolean onlyNoReply,
boolean checkForumStatus, int status, int category, int
forum) //threaddaoimpljdbc.java
```

```
...
1270. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 147:

Query Name - **Empty_Catch**

Severity - **Information**

```
1313. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, Timestamp from, Timestamp
to) //threaddaoimpljdbc.java
...
1336. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 148:

Query Name - **Empty_Catch**

Severity - **Information**

```
1597. private Collection getBeans_inForum_withSortSupport_limit_general(int forumID,
int offset, int rowsToReturn, String sort, String order, boolean enable, boolean
onlyNormalThread) //threaddaoimpljdbc.java
...
1641. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 149:

Query Name - **Empty_Catch**

Severity - **Information**

```
137. public void create(int memberID, int categoryID, int
forumID, //watchdaoimpljdbc.java
...
167. } catch(ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 150:

Query Name - **Empty_Catch**

Severity - **Information**

```
83. public void contextInitialized(ServletContextEvent
event) //mvnforumlifecycleserviceimpldefault.java
...
122. } catch (ObjectNotFoundException ex) {
```

[Table of Contents] [Description]

Path 151:

Query Name - **Empty_Catch**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
      ...
687. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 152:

Query Name - **Empty_Catch**

Severity - **Information**

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
      ...
692. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 153:

Query Name - **Empty_Catch**

Severity - **Information**

```
755. public void prepareListAttachments(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
      ...
767. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 154:

Query Name - **Empty_Catch**

Severity - **Information**

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
      ...
109. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 155:

Query Name - **Empty_Catch**

Severity - **Information**

```
101. public void process(HttpServletRequest request, HttpServletResponse
response) //forumuserservlet.java
      ...
```

```
202. } catch (AuthenticationException e) {
```

[Table of Contents] [Description]

Path 156:

Query Name - **Empty_Catch**

Severity - **Information**

```
1174. public void prepareListMembers_forPublic(GenericRequest
request) //memberwebhandler.java
      ...
1206. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 157:

Query Name - **Empty_Catch**

Severity - **Information**

```
1224. public void getAvatar(HttpServletRequest request, HttpServletResponse
response) //memberwebhandler.java
      ...
1307. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 158:

Query Name - **Empty_Catch**

Severity - **Information**

```
70. public void prepareAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
95. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 159:

Query Name - **Empty_Catch**

Severity - **Information**

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
      ...
492. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 160:

Query Name - **Empty_Catch**

Severity - **Information**

```
514. public void prepareList(GenericRequest request) //messagewebhandler.java
        ...
552. } catch (BadInputException ex) {
```

[Table of Contents] [Description]

Path 161:

Query Name - **Empty_Catch**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
        ...
496. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 162:

Query Name - **Empty_Catch**

Severity - **Information**

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
        ...
501. } catch (IOException ex) { }
```

[Table of Contents] [Description]

Path 163:

Query Name - **Empty_Catch**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
        ...
115. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 164:

Query Name - **Empty_Catch**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
        ...
457. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 165:

Query Name - **Empty_Catch**

Severity - **Information**

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
486. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 166:

Query Name - **Empty_Catch**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
      ...
877. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 167:

Query Name - **Empty_Catch**

Severity - **Information**

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
      ...
906. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 168:

Query Name - **Empty_Catch**

Severity - **Information**

```
1054. private void deletePost(GenericRequest request, PostBean
postBean) //postwebhandler.java
      ...
1130. } catch (ObjectNotFoundException ex) {
```

[Table of Contents] [Description]

Path 169:

Query Name - **Empty_Catch**

Severity - **Information**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
1270. } catch (BadInputException ex) {
```

[Table of Contents] [Description]

Path 170:

Query Name - **Empty_Catch**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
      ...
1392. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 171:

Query Name - **Empty_Catch**

Severity - **Information**

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
      ...
1411. } catch (ObjectNotFoundException e) {
```

[Table of Contents] [Description]

Path 172:

Query Name - **Empty_Catch**

Severity - **Information**

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
     ...
856. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 173:

Query Name - **Empty_Catch**

Severity - **Information**

```
920. public void prepareListRecentThreads_limit(GenericRequest
request,GenericResponse response) //threadwebhandler.java
     ...
939. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 174:

Query Name - **Empty_Catch**

Severity - **Information**

```
973. public void prepareListRecentDisabledThreads_limit(GenericRequest
request) //threadwebhandler.java
     ...
995. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 175:

Query Name - **Empty_Catch**

Severity - **Information**

```
1057. public void prepareListRecentEnableThreadsWithPendingPosts_limit
(GenericRequest request) //threadwebhandler.java
     ...
1079. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 176:

Query Name - **Empty_Catch**

Severity - **Information**

```
1107. public void prepareListEnableThreadsWithPendingPosts_inForum_limit
(GenericRequest request) //threadwebhandler.java
     ...
1133. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 177:

Query Name - **Empty_Catch**

Severity - **Information**

```
1197. public void prepareModeratePendingThreads_inForum_limit(GenericRequest
request) //threadwebhandler.java
     ...
1226. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 178:

Query Name - **Empty_Catch**

Severity - **Information**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
     ...
1303. } catch (BadInputException ex) {
```

[Table of Contents] [Description]

Path 179:

Query Name - **Empty_Catch**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1435. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 180:

Query Name - **Empty_Catch**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1454. } catch (BadInputException ex1) { }
```

[Table of Contents] [Description]

Path 181:

Query Name - **Empty_Catch**

Severity - **Information**

```
1487. public void prepareListUnansweredThreads(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
1505. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 182:

Query Name - **Empty_Catch**

Severity - **Information**

```
351. public void processAdd(GenericRequest request) //watchwebhandler.java
     ...
410. } catch (DuplicateKeyException ex) {
```

[Table of Contents] [Description]

Path 183:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //changememberstatussuccess.jsp
     ...
52. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 184:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //configindex.jsp
     ...
121. } catch (Exception e) {}
```

[Table of Contents] [Description]

Path 185:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //addmessage.jsp
     ...
61. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 186:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //addpost.jsp
     ...
68. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 187:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //listattachments.jsp
     ...
148. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 188:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //printthread.jsp
     ...
92. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 189:

Query Name - **Empty_Catch**

Severity - **Information**

```
1. <%-- //viewthread.jsp
       ...
325. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 190:

Query Name - **Empty_Catch**

Severity - **Information**

```
204. public synchronized void load(InputStream input, String
enc) //basepropertiesconfiguration.java
       ...
215. catch (UnsupportedEncodingException e)
```

[Table of Contents] [Description]

Path 191:

Query Name - **Empty_Catch**

Severity - **Information**

```
272. synchronized void freeConnection(Connection con) //dbconnectionmanager.java
       ...
284. } catch (SQLException ex) { }
```

[Table of Contents] [Description]

Path 192:

Query Name - **Empty_Catch**

Severity - **Information**

```
352. Connection getConnection(long timeout) //dbconnectionmanager.java
       ...
366. } catch (InterruptedException e) {}
```

[Table of Contents] [Description]

Path 193:

Query Name - **Empty_Catch**

Severity - **Information**

```
238. public static Connection getConnection() throws SQLException //dbutils.java
       ...
251. } catch (Exception ex) { }
```

[Table of Contents] [Description]

Path 194:

Query Name - **Empty_Catch**

Severity - **Information**

```
329. public static void resetStatement(Statement statement) //dbutils.java
       ...
339. } catch (SQLException sqle) {
```

[Table of Contents] [Description]

Path 195:

Query Name - **Empty_Catch**

Severity - **Information**

```
162. public static Connection getConnection() throws SQLException //dbutils2.java
       ...
175. } catch (Exception ex) { }
```

[Table of Contents] [Description]

Path 196:

Query Name - **Empty_Catch**

Severity - **Information**

```
45. public static String filter(String input) //htmlnewlinefilter.java
      ...
59. } catch (IOException ex) {}
```

[Table of Contents] [Description]

Path 197:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
76. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 198:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
81. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 199:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
86. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 200:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
91. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 201:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
96. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 202:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
101. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 203:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
106. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 204:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
111. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 205:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
116. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 206:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
121. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 207:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
126. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 208:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
131. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 209:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
136. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 210:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
141. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 211:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
146. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 212:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
151. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 213:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
156. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 214:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
161. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 215:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
166. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 216:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
171. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 217:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
176. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 218:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
181. } catch (ClassNotFoundException ex) {}
```

[Table of Contents] [Description]

Path 219:

Query Name - **Empty_Catch**

Severity - **Information**

```
72. public LibInfo() //libinfo.java
      ...
190. } catch (Throwable ex) {}
```

[Table of Contents] [Description]

Path 220:

Query Name - **Empty_Catch**

Severity - **Information**

```
81. public void contextDestroyed(ServletContextEvent
event) //mvncorelifecycleserviceimpldefault.java
      ...
90. } catch (CacheException e) {
```

[Table of Contents] [Description]

Path 221:

Query Name - **Empty_Catch**

Severity - **Information**

```
195. public void finishResponse() //compressionservletresponsewrapper.java
      ...
203. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 222:

Query Name - **Empty_Catch**

Severity - **Information**

```
53. public JpegEncoder(Image image, int quality, OutputStream
out) //jpegencoder.java
      ...
60. catch (InterruptedException e) {
```

[Table of Contents] [Description]

Path 223:

Query Name - **Empty_Catch**

Severity - **Information**

```
1264. private void getYCCArray() //jpegencoder.java
```

```
...
1304. catch (Exception e)
```

[Table of Contents] [Description]

Path 224:

Query Name - **Empty_Catch**

Severity - **Information**

```
1264. private void getYCCArray() //jpegencoder.java
...
1310. catch (InterruptedException e)
```

[Table of Contents] [Description]

Path 225:

Query Name - **Empty_Catch**

Severity - **Information**

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
...
443. } catch (IOException ex) {}
```

[Table of Contents] [Description]

Path 226:

Query Name - **Empty_Catch**

Severity - **Information**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
...
482. } catch (IOException ex) {}
```

[Table of Contents] [Description]

Path 227:

Query Name - **Empty_Catch**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
linesToReturn) //fileutil.java
...
524. } catch (IOException ex) {
```

[Table of Contents] [Description]

Path 228:

Query Name - **Empty_Catch**

Severity - **Information**

```
491. public static String[] getLastLines(File file, int
linesToReturn) //fileutil.java
     ...
553. } catch (IOException ex) {
```

[Table of Contents] [Description]

Path 229:

Query Name - **Empty_Catch**

Severity - **Information**

```
341. public static java.sql.Date getParameterDateSafe(GenericRequest request, String
paramDay, String paramMonth, String paramYear) //genericparamutil.java
     ...
351. } catch (BadInputException e) {
```

[Table of Contents] [Description]

Path 230:

Query Name - **Empty_Catch**

Severity - **Information**

```
647. public static boolean isClassAccessible(String className) //imageutil.java
     ...
653. } catch (ClassNotFoundException e) {} catch (LinkageError error) {} //
Thrown by some AWT classes which require awt library in static initializer.
```

[Table of Contents] [Description]

Path 231:

Query Name - **Empty_Catch**

Severity - **Information**

```
647. public static boolean isClassAccessible(String className) //imageutil.java
     ...
653. } catch (ClassNotFoundException e) {} catch (LinkageError error) {} //
Thrown by some AWT classes which require awt library in static initializer.
```

[Table of Contents] [Description]

Path 232:

Query Name - **Empty_Catch**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
     ...
355. } catch (MessagingException ex) {}
```

[Table of Contents] [Description]

Path 233:

Query Name - **Empty_Catch**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
       ...
392. } catch (MessagingException ex) { }
```

[Table of Contents] [Description]

Path 234:

Query Name - **Empty_Catch**

Severity - **Information**

```
519. protected Map /* String, String */ parseHeaders(String
headerPart) //fileuploadbase.java
       ...
558. } catch (IndexOutOfBoundsException e) {
```

[Table of Contents] [Description]

Path 235:

Query Name - **Empty_Catch**

Severity - **Information**

```
295. public byte[] get() //diskfileitem.java
       ...
315. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 236:

Query Name - **Empty_Catch**

Severity - **Information**

```
386. public void write(File file) throws Exception //diskfileitem.java
       ...
420. } catch (IOException e) {
```

[Table of Contents] [Description]

Path 237:

Query Name - **Empty_Catch**

Severity - **Information**

```
386. public void write(File file) throws Exception //diskfileitem.java
       ...
427. } catch (IOException e) {
```

[Table of Contents] [Description]

Failure to Account for Default Case in Switch

CWE ID	478
Description	The failure to account for the default case in switch statements may lead to complex logical errors and may aid in other, unexpected security-related conditions.
Common Consequences	Undefined: Depending on the logical circumstances involved, any consequences may result: e.g., issues of confidentiality, authentication, authorization, availability, integrity, accountability, or non-repudiation.
Potential Mitigations	Implementation: Ensure that there are no unaccounted for cases, when adjusting flow or values based on the value of a given variable. In switch statements, this can be accomplished through the use of the default label.

Demonstrative Examples	In general, a safe switch statement has this form:
	
	switch (value) {
	  case 'A':
	  printf("A!\n");
	  break;
	  case 'B':
	  printf("B!\n");
	  break;
	  default:
	  printf("Neither A nor B\n");
	}
	
	This is because the assumption cannot be made that all possible cases are accounted for. A good practice is to reserve the default case for error handling.

Context Notes	This flaw represents a common problem in software development, in which not all possible values for a variable are considered or handled by a given process. Because of this, further decisions are made based on poor information, and cascading failure results. This cascading failure may result in any number of security issues, and constitutes a significant failure in the system. In the case of switch style statements, the very simple act of creating a default case can mitigate this situation, if done correctly. Often however, the default cause is used simply to represent an assumed option, as opposed to working as a sanity check. This is poor practice and in some cases is as bad as omitting a default case entirely.
Node Relationships	ChildOf - Code Quality (398)
Source Taxonomies	CLASP - Failure to account for default case in switch
Applicable Platforms	C C++ Java

Path 1:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //attachmentlistimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 2:

Query Name - **No_Default_Case**

Severity - **Information**

```
184. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmentlistimpl.java
      ...
190. switch (state) {
```

[Table of Contents] [Description]

Path 3:

Query Name - **No_Default_Case**

Severity - **Information**

```
210. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmentlistimpl.java
      ...
216. switch (state) {
```

[Table of Contents] [Description]

Path 4:

Query Name - **No_Default_Case**

Severity - **Information**

```
229. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmentlistimpl.java
      ...
235. switch (state) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **No_Default_Case**

Severity - **Information**

```
248. public void handleText(final java.lang.String value) //attachmentlistimpl.java
     ...
255. switch (state) {
```

[Table of Contents] [Description]

Path 6:

Query Name - **No_Default_Case**

Severity - **Information**

```
380. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //attachmenttypeimpl.java
     ...
386. switch (state) {
```

[Table of Contents] [Description]

Path 7:

Query Name - **No_Default_Case**

Severity - **Information**

```
481. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmenttypeimpl.java
     ...
487. switch (state) {
```

[Table of Contents] [Description]

Path 8:

Query Name - **No_Default_Case**

Severity - **Information**

```
598. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmenttypeimpl.java
     ...
604. switch (state) {
```

[Table of Contents] [Description]

Path 9:

Query Name - **No_Default_Case**

Severity - **Information**

```
638. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //attachmenttypeimpl.java
     ...
644. switch (state) {
```

[Table of Contents] [Description]

Path 10:

Query Name - **No_Default_Case**

Severity - **Information**

```
678. public void handleText(final java.lang.String value) //attachmenttypeimpl.java
      ...
685. switch (state) {
```

[Table of Contents] [Description]

Path 11:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //categorylisttypeimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 12:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorylisttypeimpl.java
      ...
186. switch (state) {
```

[Table of Contents] [Description]

Path 13:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorylisttypeimpl.java
      ...
212. switch (state) {
```

[Table of Contents] [Description]

Path 14:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorylisttypeimpl.java
```

```
...
231. switch (state) {
```

[Table of Contents] [Description]

Path 15:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //categorylisttypeimpl.java
...
251. switch (state) {
```

[Table of Contents] [Description]

Path 16:

Query Name - **No_Default_Case**

Severity - **Information**

```
356. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //categorytypeimpl.java
...
362. switch (state) {
```

[Table of Contents] [Description]

Path 17:

Query Name - **No_Default_Case**

Severity - **Information**

```
472. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorytypeimpl.java
...
478. switch (state) {
```

[Table of Contents] [Description]

Path 18:

Query Name - **No_Default_Case**

Severity - **Information**

```
594. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorytypeimpl.java
...
600. switch (state) {
```

[Table of Contents] [Description]

Path 19:

Query Name - **No_Default_Case**

Severity - **Information**

```
646. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorytypeimpl.java
      |...
      |652. switch (state) {
```

[Table of Contents] [Description]

Path 20:

Query Name - **No_Default_Case**

Severity - **Information**

```
698. public void handleText(final java.lang.String value) //categorytypeimpl.java
      |...
      |705. switch (state) {
```

[Table of Contents] [Description]

Path 21:

Query Name - **No_Default_Case**

Severity - **Information**

```
163. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //categorywatchlistimpl.java
      |...
      |169. switch (state) {
```

[Table of Contents] [Description]

Path 22:

Query Name - **No_Default_Case**

Severity - **Information**

```
192. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorywatchlistimpl.java
      |...
      |198. switch (state) {
```

[Table of Contents] [Description]

Path 23:

Query Name - **No_Default_Case**

Severity - **Information**

```
218. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorywatchlistimpl.java
      |...
```

```
224. switch (state) {
```

[Table of Contents] [Description]

Path 24:

Query Name - **No_Default_Case**

Severity - **Information**

```
237. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //categorywatchlistimpl.java
...
243. switch (state) {
```

[Table of Contents] [Description]

Path 25:

Query Name - **No_Default_Case**

Severity - **Information**

```
256. public void handleText(final java.lang.String
value) //categorywatchlistimpl.java
...
263. switch (state) {
```

[Table of Contents] [Description]

Path 26:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //favoritethreadlistimpl.java
...
151. switch (state) {
```

[Table of Contents] [Description]

Path 27:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadlistimpl.java
...
186. switch (state) {
```

[Table of Contents] [Description]

Path 28:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadlistimpl.java
     |...
     |212. switch (state) {
```

[Table of Contents] [Description]

Path 29:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadlistimpl.java
     |...
     |231. switch (state) {
```

[Table of Contents] [Description]

Path 30:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //favoritethreadlistimpl.java
     |...
     |251. switch (state) {
```

[Table of Contents] [Description]

Path 31:

Query Name - **No_Default_Case**

Severity - **Information**

```
241. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //favoritethreadtypeimpl.java
     |...
     |247. switch (state) {
```

[Table of Contents] [Description]

Path 32:

Query Name - **No_Default_Case**

Severity - **Information**

```
296. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadtypeimpl.java
     |
```

```
...
302. switch (state) {
```

[Table of Contents] [Description]

Path 33:

Query Name - **No_Default_Case**

Severity - **Information**

```
359. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadtypeimpl.java
...
365. switch (state) {
```

[Table of Contents] [Description]

Path 34:

Query Name - **No_Default_Case**

Severity - **Information**

```
387. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //favoritethreadtypeimpl.java
...
393. switch (state) {
```

[Table of Contents] [Description]

Path 35:

Query Name - **No_Default_Case**

Severity - **Information**

```
415. public void handleText(final java.lang.String
value) //favoritethreadtypeimpl.java
...
422. switch (state) {
```

[Table of Contents] [Description]

Path 36:

Query Name - **No_Default_Case**

Severity - **Information**

```
144. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //forumlistimpl.java
...
150. switch (state) {
```

[Table of Contents] [Description]

Path 37:

Query Name - **No_Default_Case**

Severity - **Information**

```
183. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumlistimpl.java
      ...
189. switch (state) {
```

[Table of Contents] [Description]

Path 38:

Query Name - **No_Default_Case**

Severity - **Information**

```
209. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumlistimpl.java
      ...
215. switch (state) {
```

[Table of Contents] [Description]

Path 39:

Query Name - **No_Default_Case**

Severity - **Information**

```
228. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumlistimpl.java
      ...
234. switch (state) {
```

[Table of Contents] [Description]

Path 40:

Query Name - **No_Default_Case**

Severity - **Information**

```
247. public void handleText(final java.lang.String value) //forumlistimpl.java
      ...
254. switch (state) {
```

[Table of Contents] [Description]

Path 41:

Query Name - **No_Default_Case**

Severity - **Information**

```
566. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //forumtypeimpl.java
      ...
572. switch (state) {
```

[Table of Contents] [Description]

Path 42:

Query Name - **No_Default_Case**

Severity - **Information**

```
761. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumtypeimpl.java
      ...
767. switch (state) {
```

[Table of Contents] [Description]

Path 43:

Query Name - **No_Default_Case**

Severity - **Information**

```
976. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumtypeimpl.java
      ...
982. switch (state) {
```

[Table of Contents] [Description]

Path 44:

Query Name - **No_Default_Case**

Severity - **Information**

```
1058. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumtypeimpl.java
      ...
1064. switch (state) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **No_Default_Case**

Severity - **Information**

```
1140. public void handleText(final java.lang.String value) //forumtypeimpl.java
      ...
1147. switch (state) {
```

[Table of Contents] [Description]

Path 46:

Query Name - **No_Default_Case**

Severity - **Information**

```
163. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
```

```
java.lang.String ___qname, org.xml.sax.Attributes __atts) //forumwatchlistimpl.java
      ...
169. switch (state) {
```

[Table of Contents] [Description]

Path 47:

Query Name - **No_Default_Case**

Severity - **Information**

```
192. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumwatchlistimpl.java
      ...
198. switch (state) {
```

[Table of Contents] [Description]

Path 48:

Query Name - **No_Default_Case**

Severity - **Information**

```
218. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumwatchlistimpl.java
      ...
224. switch (state) {
```

[Table of Contents] [Description]

Path 49:

Query Name - **No_Default_Case**

Severity - **Information**

```
237. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //forumwatchlistimpl.java
      ...
243. switch (state) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **No_Default_Case**

Severity - **Information**

```
256. public void handleText(final java.lang.String value) //forumwatchlistimpl.java
      ...
263. switch (state) {
```

[Table of Contents] [Description]

Path 51:

Query Name - **No_Default_Case**

Severity - **Information**

```
182. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //globalpermissionlistimpl.java
      |...
      |188. switch (state) {
```

[Table of Contents] [Description]

Path 52:

Query Name - **No_Default_Case**

Severity - **Information**

```
211. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalpermissionlistimpl.java
      |...
      |217. switch (state) {
```

[Table of Contents] [Description]

Path 53:

Query Name - **No_Default_Case**

Severity - **Information**

```
237. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalpermissionlistimpl.java
      |...
      |243. switch (state) {
```

[Table of Contents] [Description]

Path 54:

Query Name - **No_Default_Case**

Severity - **Information**

```
256. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalpermissionlistimpl.java
      |...
      |262. switch (state) {
```

[Table of Contents] [Description]

Path 55:

Query Name - **No_Default_Case**

Severity - **Information**

```
275. public void handleText(final java.lang.String
value) //globalpermissionlistimpl.java
      |
```

```
...
282. switch (state) {
```

[Table of Contents] [Description]

Path 56:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //globalwatchlistimpl.java
...
151. switch (state) {
```

[Table of Contents] [Description]

Path 57:

Query Name - **No_Default_Case**

Severity - **Information**

```
201. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchlistimpl.java
...
207. switch (state) {
```

[Table of Contents] [Description]

Path 58:

Query Name - **No_Default_Case**

Severity - **Information**

```
230. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchlistimpl.java
...
236. switch (state) {
```

[Table of Contents] [Description]

Path 59:

Query Name - **No_Default_Case**

Severity - **Information**

```
252. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchlistimpl.java
...
258. switch (state) {
```

[Table of Contents] [Description]

Path 60:

Query Name - **No_Default_Case**

Severity - **Information**

```
274. public void handleText(final java.lang.String value) //globalwatchlistimpl.java
      ...
281. switch (state) {
```

[Table of Contents] [Description]

Path 61:

Query Name - **No_Default_Case**

Severity - **Information**

```
265. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //globalwatchtypeimpl.java
      ...
271. switch (state) {
```

[Table of Contents] [Description]

Path 62:

Query Name - **No_Default_Case**

Severity - **Information**

```
329. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchtypeimpl.java
      ...
335. switch (state) {
```

[Table of Contents] [Description]

Path 63:

Query Name - **No_Default_Case**

Severity - **Information**

```
405. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchtypeimpl.java
      ...
411. switch (state) {
```

[Table of Contents] [Description]

Path 64:

Query Name - **No_Default_Case**

Severity - **Information**

```
439. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //globalwatchtypeimpl.java
      ...
445. switch (state) {
```

[Table of Contents] [Description]

Path 65:

Query Name - **No_Default_Case**

Severity - **Information**

```
473. public void handleText(final java.lang.String value) //globalwatchtypeimpl.java
      ...
480. switch (state) {
```

[Table of Contents] [Description]

Path 66:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //groupforumpermissionlistimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 67:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissionlistimpl.java
      ...
186. switch (state) {
```

[Table of Contents] [Description]

Path 68:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissionlistimpl.java
      ...
212. switch (state) {
```

[Table of Contents] [Description]

Path 69:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissionlistimpl.java
     ...
231. switch (state) {
```

[Table of Contents] [Description]

Path 70:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //groupforumpermissionlistimpl.java
     ...
251. switch (state) {
```

[Table of Contents] [Description]

Path 71:

Query Name - **No_Default_Case**

Severity - **Information**

```
177. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //groupforumpermissiontypeimpl.java
     ...
183. switch (state) {
```

[Table of Contents] [Description]

Path 72:

Query Name - **No_Default_Case**

Severity - **Information**

```
207. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissiontypeimpl.java
     ...
213. switch (state) {
```

[Table of Contents] [Description]

Path 73:

Query Name - **No_Default_Case**

Severity - **Information**

```
237. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissiontypeimpl.java
     ...
243. switch (state) {
```

[Table of Contents] [Description]

Path 74:

Query Name - **No_Default_Case**

Severity - **Information**

```
253. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupforumpermissiontypeimpl.java
      ...
259. switch (state) {
```

[Table of Contents] [Description]

Path 75:

Query Name - **No_Default_Case**

Severity - **Information**

```
269. public void handleText(final java.lang.String
value) //groupforumpermissiontypeimpl.java
      ...
276. switch (state) {
```

[Table of Contents] [Description]

Path 76:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //groupmemberlistimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 77:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmemberlistimpl.java
      ...
186. switch (state) {
```

[Table of Contents] [Description]

Path 78:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmemberlistimpl.java
      ...
212. switch (state) {
```

[Table of Contents] [Description]

Path 79:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmemberlistimpl.java
      ...
231. switch (state) {
```

[Table of Contents] [Description]

Path 80:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String value) //groupmemberlistimpl.java
      ...
251. switch (state) {
```

[Table of Contents] [Description]

Path 81:

Query Name - **No_Default_Case**

Severity - **Information**

```
214. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //groupmembertypeimpl.java
      ...
220. switch (state) {
```

[Table of Contents] [Description]

Path 82:

Query Name - **No_Default_Case**

Severity - **Information**

```
261. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmembertypeimpl.java
      ...
267. switch (state) {
```

[Table of Contents] [Description]

Path 83:

Query Name - **No_Default_Case**

Severity - **Information**

```
314. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmembertypeimpl.java
     ...
320. switch (state) {
```

[Table of Contents] [Description]

Path 84:

Query Name - **No_Default_Case**

Severity - **Information**

```
339. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //groupmembertypeimpl.java
     ...
345. switch (state) {
```

[Table of Contents] [Description]

Path 85:

Query Name - **No_Default_Case**

Severity - **Information**

```
364. public void handleText(final java.lang.String value) //groupmembertypeimpl.java
     ...
371. switch (state) {
```

[Table of Contents] [Description]

Path 86:

Query Name - **No_Default_Case**

Severity - **Information**

```
308. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //grouptypeimpl.java
     ...
314. switch (state) {
```

[Table of Contents] [Description]

Path 87:

Query Name - **No_Default_Case**

Severity - **Information**

```
401. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //grouptypeimpl.java
     ...
```

```
407. switch (state) {
```

[Table of Contents] [Description]

Path 88:

Query Name - **No_Default_Case**

Severity - **Information**

```
500. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //grouptypeimpl.java
...
506. switch (state) {
```

[Table of Contents] [Description]

Path 89:

Query Name - **No_Default_Case**

Severity - **Information**

```
543. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //grouptypeimpl.java
...
549. switch (state) {
```

[Table of Contents] [Description]

Path 90:

Query Name - **No_Default_Case**

Severity - **Information**

```
586. public void handleText(final java.lang.String value) //grouptypeimpl.java
...
593. switch (state) {
```

[Table of Contents] [Description]

Path 91:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //memberforumpermissionlistimpl.java
...
151. switch (state) {
```

[Table of Contents] [Description]

Path 92:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissionlistimpl.java
     ...
186. switch (state) {
```

[Table of Contents] [Description]

Path 93:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissionlistimpl.java
     ...
212. switch (state) {
```

[Table of Contents] [Description]

Path 94:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissionlistimpl.java
     ...
231. switch (state) {
```

[Table of Contents] [Description]

Path 95:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String
value) //memberforumpermissionlistimpl.java
     ...
251. switch (state) {
```

[Table of Contents] [Description]

Path 96:

Query Name - **No_Default_Case**

Severity - **Information**

```
177. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //memberforumpermissiontypeimpl.java
     ...
183. switch (state) {
```

[Table of Contents] [Description]

Path 97:

Query Name - **No_Default_Case**

Severity - **Information**

```
207. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissiontypeimpl.java
      ...
213. switch (state) {
```

[Table of Contents] [Description]

Path 98:

Query Name - **No_Default_Case**

Severity - **Information**

```
237. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissiontypeimpl.java
      ...
243. switch (state) {
```

[Table of Contents] [Description]

Path 99:

Query Name - **No_Default_Case**

Severity - **Information**

```
253. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //memberforumpermissiontypeimpl.java
      ...
259. switch (state) {
```

[Table of Contents] [Description]

Path 100:

Query Name - **No_Default_Case**

Severity - **Information**

```
269. public void handleText(final java.lang.String
value) //memberforumpermissiontypeimpl.java
      ...
276. switch (state) {
```

[Table of Contents] [Description]

Path 101:

Query Name - **No_Default_Case**

Severity - **Information**

```
1343. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //membertypeimpl.java
      ...
1349. switch (state) {
```

[Table of Contents] [Description]

Path 102:

Query Name - **No_Default_Case**

Severity - **Information**

```
1812. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //membertypeimpl.java
      ...
1818. switch (state) {
```

[Table of Contents] [Description]

Path 103:

Query Name - **No_Default_Case**

Severity - **Information**

```
2359. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //membertypeimpl.java
      ...
2365. switch (state) {
```

[Table of Contents] [Description]

Path 104:

Query Name - **No_Default_Case**

Severity - **Information**

```
2528. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //membertypeimpl.java
      ...
2534. switch (state) {
```

[Table of Contents] [Description]

Path 105:

Query Name - **No_Default_Case**

Severity - **Information**

```
2697. public void handleText(final java.lang.String value) //membertypeimpl.java
      ...
2704. switch (state) {
```

[Table of Contents] [Description]

Path 106:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //messagefolderlistimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 107:

Query Name - **No_Default_Case**

Severity - **Information**

```
180. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefolderlistimpl.java
      ...
186. switch (state) {
```

[Table of Contents] [Description]

Path 108:

Query Name - **No_Default_Case**

Severity - **Information**

```
206. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefolderlistimpl.java
      ...
212. switch (state) {
```

[Table of Contents] [Description]

Path 109:

Query Name - **No_Default_Case**

Severity - **Information**

```
225. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefolderlistimpl.java
      ...
231. switch (state) {
```

[Table of Contents] [Description]

Path 110:

Query Name - **No_Default_Case**

Severity - **Information**

```
244. public void handleText(final java.lang.String
```

```
value) //messagefolderlistimpl.java
      ...
      251. switch (state) {
```

[Table of Contents] [Description]

Path 111:

Query Name - **No_Default_Case**

Severity - **Information**

```
241. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes
__atts) //messagefoldertypeimpl.java
      ...
      247. switch (state) {
```

[Table of Contents] [Description]

Path 112:

Query Name - **No_Default_Case**

Severity - **Information**

```
303. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefoldertypeimpl.java
      ...
      309. switch (state) {
```

[Table of Contents] [Description]

Path 113:

Query Name - **No_Default_Case**

Severity - **Information**

```
369. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefoldertypeimpl.java
      ...
      375. switch (state) {
```

[Table of Contents] [Description]

Path 114:

Query Name - **No_Default_Case**

Severity - **Information**

```
400. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagefoldertypeimpl.java
      ...
      406. switch (state) {
```

[Table of Contents] [Description]

Path 115:

Query Name - **No_Default_Case**

Severity - **Information**

```
431. public void handleText(final java.lang.String
value) //messagefoldertypeimpl.java
      ...
438. switch (state) {
```

[Table of Contents] [Description]

Path 116:

Query Name - **No_Default_Case**

Severity - **Information**

```
144. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //messagelistimpl.java
      ...
150. switch (state) {
```

[Table of Contents] [Description]

Path 117:

Query Name - **No_Default_Case**

Severity - **Information**

```
240. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagelistimpl.java
      ...
246. switch (state) {
```

[Table of Contents] [Description]

Path 118:

Query Name - **No_Default_Case**

Severity - **Information**

```
269. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagelistimpl.java
      ...
275. switch (state) {
```

[Table of Contents] [Description]

Path 119:

Query Name - **No_Default_Case**

Severity - **Information**

```
291. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
```

```
java.lang.String ___qname) //messagelistimpl.java
      |...
      |297. switch (state) {
```

[Table of Contents] [Description]

Path 120:

Query Name - **No_Default_Case**

Severity - **Information**

```
313. public void handleText(final java.lang.String value) //messagelistimpl.java
      |...
      |320. switch (state) {
```

[Table of Contents] [Description]

Path 121:

Query Name - **No_Default_Case**

Severity - **Information**

```
493. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //messagetypeimpl.java
      |...
      |499. switch (state) {
```

[Table of Contents] [Description]

Path 122:

Query Name - **No_Default_Case**

Severity - **Information**

```
637. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagetypeimpl.java
      |...
      |643. switch (state) {
```

[Table of Contents] [Description]

Path 123:

Query Name - **No_Default_Case**

Severity - **Information**

```
813. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagetypeimpl.java
      |...
      |819. switch (state) {
```

[Table of Contents] [Description]

Path 124:

Query Name - **No_Default_Case**

Severity - **Information**

```
877. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //messagetypeimpl.java
      |...
      |883. switch (state) {
```

[Table of Contents] [Description]

Path 125:

Query Name - **No_Default_Case**

Severity - **Information**

```
941. public void handleText(final java.lang.String value) //messagetypeimpl.java
      |...
      |948. switch (state) {
```

[Table of Contents] [Description]

Path 126:

Query Name - **No_Default_Case**

Severity - **Information**

```
140. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //mvnforumimpl.java
      |...
      |146. switch (state) {
```

[Table of Contents] [Description]

Path 127:

Query Name - **No_Default_Case**

Severity - **Information**

```
182. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumimpl.java
      |...
      |188. switch (state) {
```

[Table of Contents] [Description]

Path 128:

Query Name - **No_Default_Case**

Severity - **Information**

```
208. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumimpl.java
      |...
      |214. switch (state) {
```

[Table of Contents] [Description]

Path 129:

Query Name - **No_Default_Case**

Severity - **Information**

```
227. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumimpl.java
     |...
     |233. switch (state) {
```

[Table of Contents] [Description]

Path 130:

Query Name - **No_Default_Case**

Severity - **Information**

```
246. public void handleText(final java.lang.String value) //mvnforumimpl.java
     |...
     |253. switch (state) {
```

[Table of Contents] [Description]

Path 131:

Query Name - **No_Default_Case**

Severity - **Information**

```
309. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //mvnforumtypeimpl.java
     |...
     |315. switch (state) {
```

[Table of Contents] [Description]

Path 132:

Query Name - **No_Default_Case**

Severity - **Information**

```
348. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
     |...
     |354. switch (state) {
```

[Table of Contents] [Description]

Path 133:

Query Name - **No_Default_Case**

Severity - **Information**

```
374. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
```

```
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
380. switch (state) {
```

[Table of Contents] [Description]

Path 134:

Query Name - **No_Default_Case**

Severity - **Information**

```
393. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
399. switch (state) {
```

[Table of Contents] [Description]

Path 135:

Query Name - **No_Default_Case**

Severity - **Information**

```
412. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
    ...
419. switch (state) {
```

[Table of Contents] [Description]

Path 136:

Query Name - **No_Default_Case**

Severity - **Information**

```
571. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //mvnforumtypeimpl.java
    ...
577. switch (state) {
```

[Table of Contents] [Description]

Path 137:

Query Name - **No_Default_Case**

Severity - **Information**

```
606. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
612. switch (state) {
```

[Table of Contents] [Description]

Path 138:

Query Name - **No_Default_Case**

Severity - **Information**

```
632. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
      ...
638. switch (state) {
```

[Table of Contents] [Description]

Path 139:

Query Name - **No_Default_Case**

Severity - **Information**

```
651. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
      ...
657. switch (state) {
```

[Table of Contents] [Description]

Path 140:

Query Name - **No_Default_Case**

Severity - **Information**

```
670. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
      ...
677. switch (state) {
```

[Table of Contents] [Description]

Path 141:

Query Name - **No_Default_Case**

Severity - **Information**

```
829. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //mvnforumtypeimpl.java
      ...
835. switch (state) {
```

[Table of Contents] [Description]

Path 142:

Query Name - **No_Default_Case**

Severity - **Information**

```
864. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
      ...
870. switch (state) {
```

[Table of Contents] [Description]

Path 143:

Query Name - **No_Default_Case**

Severity - **Information**

```
890. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
     ...
896. switch (state) {
```

[Table of Contents] [Description]

Path 144:

Query Name - **No_Default_Case**

Severity - **Information**

```
909. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
     ...
915. switch (state) {
```

[Table of Contents] [Description]

Path 145:

Query Name - **No_Default_Case**

Severity - **Information**

```
928. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
     ...
935. switch (state) {
```

[Table of Contents] [Description]

Path 146:

Query Name - **No_Default_Case**

Severity - **Information**

```
972. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //mvnforumtypeimpl.java
     ...
978. switch (state) {
```

[Table of Contents] [Description]

Path 147:

Query Name - **No_Default_Case**

Severity - **Information**

```
1048. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
```

```
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
1054. switch (state) {
```

[Table of Contents] [Description]

Path 148:

Query Name - **No_Default_Case**

Severity - **Information**

```
1116. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
1122. switch (state) {
```

[Table of Contents] [Description]

Path 149:

Query Name - **No_Default_Case**

Severity - **Information**

```
1156. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //mvnforumtypeimpl.java
    ...
1162. switch (state) {
```

[Table of Contents] [Description]

Path 150:

Query Name - **No_Default_Case**

Severity - **Information**

```
1196. public void handleText(final java.lang.String value) //mvnforumtypeimpl.java
    ...
1203. switch (state) {
```

[Table of Contents] [Description]

Path 151:

Query Name - **No_Default_Case**

Severity - **Information**

```
144. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //postlistimpl.java
    ...
150. switch (state) {
```

[Table of Contents] [Description]

Path 152:

Query Name - **No_Default_Case**

Severity - **Information**

```
187. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //postlistimpl.java
      ...
193. switch (state) {
```

[Table of Contents] [Description]

Path 153:

Query Name - **No_Default_Case**

Severity - **Information**

```
213. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //postlistimpl.java
      ...
219. switch (state) {
```

[Table of Contents] [Description]

Path 154:

Query Name - **No_Default_Case**

Severity - **Information**

```
232. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //postlistimpl.java
      ...
238. switch (state) {
```

[Table of Contents] [Description]

Path 155:

Query Name - **No_Default_Case**

Severity - **Information**

```
251. public void handleText(final java.lang.String value) //postlistimpl.java
      ...
258. switch (state) {
```

[Table of Contents] [Description]

Path 156:

Query Name - **No_Default_Case**

Severity - **Information**

```
490. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //posttypeimpl.java
      ...
496. switch (state) {
```

[Table of Contents] [Description]

Path 157:

Query Name - **No_Default_Case**

Severity - **Information**

```
646. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //posttypeimpl.java
      |...
      |652. switch (state) {
```

[Table of Contents] [Description]

Path 158:

Query Name - **No_Default_Case**

Severity - **Information**

```
822. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //posttypeimpl.java
      |...
      |828. switch (state) {
```

[Table of Contents] [Description]

Path 159:

Query Name - **No_Default_Case**

Severity - **Information**

```
886. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //posttypeimpl.java
      |...
      |892. switch (state) {
```

[Table of Contents] [Description]

Path 160:

Query Name - **No_Default_Case**

Severity - **Information**

```
950. public void handleText(final java.lang.String value) //posttypeimpl.java
      |...
      |957. switch (state) {
```

[Table of Contents] [Description]

Path 161:

Query Name - **No_Default_Case**

Severity - **Information**

```
271. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
```

```
java.lang.String ___qname, org.xml.sax.Attributes __atts) //ranktypeimpl.java
    ...
277. switch (state) {
```

[Table of Contents] [Description]

Path 162:

Query Name - **No_Default_Case**

Severity - **Information**

```
333. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //ranktypeimpl.java
    ...
339. switch (state) {
```

[Table of Contents] [Description]

Path 163:

Query Name - **No_Default_Case**

Severity - **Information**

```
403. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //ranktypeimpl.java
    ...
409. switch (state) {
```

[Table of Contents] [Description]

Path 164:

Query Name - **No_Default_Case**

Severity - **Information**

```
431. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //ranktypeimpl.java
    ...
437. switch (state) {
```

[Table of Contents] [Description]

Path 165:

Query Name - **No_Default_Case**

Severity - **Information**

```
459. public void handleText(final java.lang.String value) //ranktypeimpl.java
    ...
466. switch (state) {
```

[Table of Contents] [Description]

Path 166:

Query Name - **No_Default_Case**

Severity - **Information**

```
144. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //threadlistimpl.java
      ...
150. switch (state) {
```

[Table of Contents] [Description]

Path 167:

Query Name - **No_Default_Case**

Severity - **Information**

```
187. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadlistimpl.java
      ...
193. switch (state) {
```

[Table of Contents] [Description]

Path 168:

Query Name - **No_Default_Case**

Severity - **Information**

```
213. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadlistimpl.java
      ...
219. switch (state) {
```

[Table of Contents] [Description]

Path 169:

Query Name - **No_Default_Case**

Severity - **Information**

```
232. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadlistimpl.java
      ...
238. switch (state) {
```

[Table of Contents] [Description]

Path 170:

Query Name - **No_Default_Case**

Severity - **Information**

```
251. public void handleText(final java.lang.String value) //threadlistimpl.java
      ...
258. switch (state) {
```

[Table of Contents] [Description]

Path 171:

Query Name - **No_Default_Case**

Severity - **Information**

```
565. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //threadtypeimpl.java
      ...
571. switch (state) {
```

[Table of Contents] [Description]

Path 172:

Query Name - **No_Default_Case**

Severity - **Information**

```
752. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadtypeimpl.java
      ...
758. switch (state) {
```

[Table of Contents] [Description]

Path 173:

Query Name - **No_Default_Case**

Severity - **Information**

```
961. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadtypeimpl.java
      ...
967. switch (state) {
```

[Table of Contents] [Description]

Path 174:

Query Name - **No_Default_Case**

Severity - **Information**

```
1037. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadtypeimpl.java
      ...
1043. switch (state) {
```

[Table of Contents] [Description]

Path 175:

Query Name - **No_Default_Case**

Severity - **Information**

```
1113. public void handleText(final java.lang.String value) //threadtypeimpl.java
      ...
1120. switch (state) {
```

[Table of Contents] [Description]

Path 176:

Query Name - **No_Default_Case**

Severity - **Information**

```
145. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //threadwatchlistimpl.java
      ...
151. switch (state) {
```

[Table of Contents] [Description]

Path 177:

Query Name - **No_Default_Case**

Severity - **Information**

```
205. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchlistimpl.java
      ...
211. switch (state) {
```

[Table of Contents] [Description]

Path 178:

Query Name - **No_Default_Case**

Severity - **Information**

```
234. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchlistimpl.java
      ...
240. switch (state) {
```

[Table of Contents] [Description]

Path 179:

Query Name - **No_Default_Case**

Severity - **Information**

```
256. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchlistimpl.java
      ...
262. switch (state) {
```

[Table of Contents] [Description]

Path 180:

Query Name - **No_Default_Case**

Severity - **Information**

```
278. public void handleText(final java.lang.String value) //threadwatchlistimpl.java
      ...
285. switch (state) {
```

[Table of Contents] [Description]

Path 181:

Query Name - **No_Default_Case**

Severity - **Information**

```
287. public void enterElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname, org.xml.sax.Attributes __atts) //threadwatchtypeimpl.java
      ...
293. switch (state) {
```

[Table of Contents] [Description]

Path 182:

Query Name - **No_Default_Case**

Severity - **Information**

```
359. public void leaveElement(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchtypeimpl.java
      ...
365. switch (state) {
```

[Table of Contents] [Description]

Path 183:

Query Name - **No_Default_Case**

Severity - **Information**

```
445. public void enterAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchtypeimpl.java
      ...
451. switch (state) {
```

[Table of Contents] [Description]

Path 184:

Query Name - **No_Default_Case**

Severity - **Information**

```
482. public void leaveAttribute(java.lang.String ___uri, java.lang.String ___local,
java.lang.String ___qname) //threadwatchtypeimpl.java
      ...
```

```
488. switch (state) {
```

[Table of Contents] [Description]

Path 185:

Query Name - **No_Default_Case**

Severity - **Information**

```
519. public void handleText(final java.lang.String value) //threadwatchtypeimpl.java
     ...
526. switch (state) {
```

[Table of Contents] [Description]

Path 186:

Query Name - **No_Default_Case**

Severity - **Information**

```
177. public static String getDatabaseTypeName(int databaseType) //dbutils.java
     ...
179. switch (databaseType) {
```

[Table of Contents] [Description]

Erroneous String Compare

CWE ID	597
Description	Strings should be compared with the equals() method, not == or !=

Demonstrative Examples	The following branch will never be taken.
	
	if (args[0] == STRING_CONSTANT) {
	  logger.info("miracle");
	}

Context Notes	Using == or != to compare two strings for equality actually compares two objects for equality, not their values. Chances are good that the two references will never be equal.
Node Relationships	ChildOf - Incorrect Object Comparison: Syntactic (595) ChildOf - Incorrect Object Comparison: Syntactic (595) ChildOf - String Errors (133)

Path 1:

Query Name - **Erroneous_String_Compare**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
     ...
423. if( auri=="http://www.w3.org/2001/XMLSchema-instance" && alocal=="nil" )
```

```
{
```

[Table of Contents] [Description]

Path 2:

Query Name - **Erroneous_String_Compare**

Severity - **Information**

```
373. public void pushAttributes( Attributes atts, boolean
collectTextFlag ) //saxunmarshallerhandlerimpl.java
...
423. if( auri=="http://www.w3.org/2001/XMLSchema-instance" && alocal=="nil" )
{
```

[Table of Contents] [Description]

Magic Numbers

CWE ID	10017
Description	Using literals in the code makes the code less readable and maintainable.
Likelihood of Exploit	Low
Common Consequences	Code become less readable and maintainable.
Potential Mitigations	Define constants with meaningful names and use them instead.
Applicable Platforms	All

Path 1:

Query Name - **Magic_Numbers**

Severity - **Information**

```
138. public class Migrator //migrator.java
...
142. private static int PHPBB_ADMIN_ID = 2;
```

[Table of Contents] [Description]

Path 2:

Query Name - **Magic_Numbers**

Severity - **Information**

```
138. public class Migrator //migrator.java
...
148. private static int GUEST_ID = 2;
```

[Table of Contents] [Description]

Path 3:

Query Name - **Magic_Numbers**

Severity - **Information**

```
158. public static void migrateUsers(PrintStream ostream, int
gender) //migrator.java
      ...
203. int memberPostsPerPage = 10; //Default by mvnForum.
```

[Table of Contents] [Description]

Path 4:

Query Name - **Magic_Numbers**

Severity - **Information**

```
100. public class ImportExportXML //importexportxml.java
      ...
106. private static int GROUP_ADMIN_PERMISSION = 100;
```

[Table of Contents] [Description]

Path 5:

Query Name - **Magic_Numbers**

Severity - **Information**

```
100. public class ImportExportXML //importexportxml.java
      ...
108. private static int GROUP_GUEST_PERMISSION = 109;
```

[Table of Contents] [Description]

Path 6:

Query Name - **Magic_Numbers**

Severity - **Information**

```
100. public class ImportExportXML //importexportxml.java
      ...
112. private static int GUEST_ID = 2;
```

[Table of Contents] [Description]

Path 7:

Query Name - **Magic_Numbers**

Severity - **Information**

```
492. public void exportMemberList() //importexportxml.java
      ...
736. int memberPostsPerPage = 10; //Default by mvnForum.
```

[Table of Contents] [Description]

Path 8:

Query Name - **Magic_Numbers**

Severity - **Information**

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
     ...
122. int timeZone = 7;
```

[Table of Contents] [Description]

Path 9:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
     ...
737. private static int maxAttachmentSize = 1024;
```

[Table of Contents] [Description]

Path 10:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
     ...
745. private static int maxMessageAttachmentSize = 1024;
```

[Table of Contents] [Description]

Path 11:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
     ...
1050. private static int thumbnailWidth = 100;
```

[Table of Contents] [Description]

Path 12:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
     ...
1058. private static int thumbnailHeight = 100;
```

[Table of Contents] [Description]

Path 13:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1135. private static int maxFavoriteThreads = 128;
```

[Table of Contents] [Description]

Path 14:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1143. private static int maxPrivateMessages = 128;
```

[Table of Contents] [Description]

Path 15:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1151. private static int maxHotTopics = 10;
```

[Table of Contents] [Description]

Path 16:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1159. private static int maxPostsPerHourPerIP = 20;
```

[Table of Contents] [Description]

Path 17:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1167. private static int maxPostsPerHourPerMember = 10;
```

[Table of Contents] [Description]

Path 18:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1175. private static int maxMembersPerHourPerIP = 2;
```

[Table of Contents] [Description]

Path 19:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1183. private static int maxLoginsPerHourPerIP = 5;
```

[Table of Contents] [Description]

Path 20:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1191. private static int maxHttpRequestsPerHourPerIP = 200;
```

[Table of Contents] [Description]

Path 21:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1199. private static int maxMessagesPerHourPerIP = 5;
```

[Table of Contents] [Description]

Path 22:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1212. private static int maxActiveThreads = 5;
```

[Table of Contents] [Description]

Path 23:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1220. private static int maxActiveMembers = 5;
```

[Table of Contents] [Description]

Path 24:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1239. private static int maxImportSize = 4096000;
```

[Table of Contents] [Description]

Path 25:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1294. private static int maxEditDays = 7;
```

[Table of Contents] [Description]

Path 26:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1327. private static int rowsPerPage = 10;
```

[Table of Contents] [Description]

Path 27:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1338. private static int rowsPerRSS = 15; // RSS 0.91
```

[Table of Contents] [Description]

Path 28:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1418. private static int maxCharsInShortSummary = 100;
```

[Table of Contents] [Description]

Path 29:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1427. private static int maxCharsInLongSummary = 1000;
```

[Table of Contents] [Description]

Path 30:

Query Name - **Magic_Numbers**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1436. private static int maxCharsInRSS = 500;
```

[Table of Contents] [Description]

Path 31:

Query Name - **Magic_Numbers**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
142. int sizeThreshold = 100000;// max memory used = 100K
```

[Table of Contents] [Description]

Path 32:

Query Name - **Magic_Numbers**

Severity - **Information**

```
103. public void processAdd(GenericRequest request) //memberwebhandler.java
      ...
176. int memberPostsPerPage = 10; //default for all preregistered users
```

[Table of Contents] [Description]

Path 33:

Query Name - **Magic_Numbers**

Severity - **Information**

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
91. int postsPerPage = 10;
```

[Table of Contents] [Description]

Path 34:

Query Name - **Magic_Numbers**

Severity - **Information**

```
65. class OnlineUserImpl implements OnlineUser //onlineuserimpl.java
      ...
81. private int memberPostsPerPage = 10;
```

[Table of Contents] [Description]

Path 35:

Query Name - **Magic_Numbers**

Severity - **Information**

```
65. class OnlineUserImpl implements OnlineUser //onlineuserimpl.java
      ...
82. private int memberMessagesPerPage = 10;
```

[Table of Contents] [Description]

Path 36:

Query Name - **Magic_Numbers**

Severity - **Information**

```
82. public static void addNewMember(HttpServletRequest request, String memberName,
String memberPassword, String memberEmail, Date memberBirthday) //kgutils.java
      ...
116. int memberPostsPerPage = 10; //default for all preregistered users
```

[Table of Contents] [Description]

Path 37:

Query Name - **Magic_Numbers**

Severity - **Information**

```
146. public void processAdd(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
      ...
```

```
173. int sizeThreshold = 100000;
```

[Table of Contents] [Description]

Path 38:

Query Name - **Magic_Numbers**

Severity - **Information**

```
782. public void uploadAvatar(javax.servlet.ServletConfig config, GenericRequest
request) //memberwebhandler.java
...
801. int sizeMax = 60000; // 60KB
```

[Table of Contents] [Description]

Path 39:

Query Name - **Magic_Numbers**

Severity - **Information**

```
782. public void uploadAvatar(javax.servlet.ServletConfig config, GenericRequest
request) //memberwebhandler.java
...
802. int sizeThreshold = 100000;// max memory used = 100K (more than needed)
```

[Table of Contents] [Description]

Path 40:

Query Name - **Magic_Numbers**

Severity - **Information**

```
67. public class PostUtil //postutil.java
...
71. public static int NUMBER_OF_MORE_THREADS_TO_GET = 5;
```

[Table of Contents] [Description]

Path 41:

Query Name - **Magic_Numbers**

Severity - **Information**

```
1157. public void prepareModeratePendingPosts_limit(GenericRequest
request) //postwebhandler.java
...
1191. int postsPerPage = 10000;
```

[Table of Contents] [Description]

Path 42:

Query Name - **Magic_Numbers**

Severity - **Information**

```
45. public class MVNCoreConfig //mvncoreconfig.java
      ...
109. private static int maxConnection = 20;
```

[Table of Contents] [Description]

Path 43:

Query Name - **Magic_Numbers**

Severity - **Information**

```
45. public class MVNCoreConfig //mvncoreconfig.java
      ...
118. private static int maxTimeToWait = 2000;// 2 seconds
```

[Table of Contents] [Description]

Path 44:

Query Name - **Magic_Numbers**

Severity - **Information**

```
45. public class MVNCoreConfig //mvncoreconfig.java
      ...
126. private static int minutesBetweenRefresh = 30;// 30 minutes
```

[Table of Contents] [Description]

Path 45:

Query Name - **Magic_Numbers**

Severity - **Information**

```
52. public final class DBUtils //dbutils.java
      ...
81. private static int maxTimeToWait = 2000;// 2 seconds
```

[Table of Contents] [Description]

Path 46:

Query Name - **Magic_Numbers**

Severity - **Information**

```
52. public final class DBUtils //dbutils.java
      ...
83. private static int minutesBetweenRefresh = 30;// 30 minutes
```

[Table of Contents] [Description]

Path 47:

Query Name - **Magic_Numbers**

Severity - **Information**

```
52. public final class DBUtils2 //dbutils2.java
      ...
60. private static int maxTimeToWait = 2000;// 2 seconds
```

[Table of Contents] [Description]

Path 48:

Query Name - **Magic_Numbers**

Severity - **Information**

```
52. public final class DBUtils2 //dbutils2.java
      ...
62. private static int minutesBetweenRefresh = 30;// 30 minutes
```

[Table of Contents] [Description]

Path 49:

Query Name - **Magic_Numbers**

Severity - **Information**

```
107. public final class Base64 //base64.java
      ...
147. for (int i = 26, j = 0; i <= 51; i++, j++ )
```

[Table of Contents] [Description]

Path 50:

Query Name - **Magic_Numbers**

Severity - **Information**

```
107. public final class Base64 //base64.java
      ...
150. for (int i = 52, j = 0; i <= 61; i++, j++ )
```

[Table of Contents] [Description]

Path 51:

Query Name - **Magic_Numbers**

Severity - **Information**

```
51. public class FloodControl //floodcontrol.java
      ...
56. public static int MAX_MVNCORE_ACTION_ID = 999;
```

[Table of Contents] [Description]

Path 52:

Query Name - **Magic_Numbers**

Severity - **Information**

```
87. public class CompressionFilter implements Filter //compressionfilter.java
      ...
100. private int minThreshold = 128;
```

[Table of Contents] [Description]

Path 53:

Query Name - **Magic_Numbers**

Severity - **Information**

```
343. class DCT //jpegencoder.java
      ...
348. public int N = 8;
```

[Table of Contents] [Description]

Path 54:

Query Name - **Magic_Numbers**

Severity - **Information**

```
343. class DCT //jpegencoder.java
      ...
353. public int QUALITY = 80;
```

[Table of Contents] [Description]

Path 55:

Query Name - **Magic_Numbers**

Severity - **Information**

```
1207. class JpegInfo //jpegencoder.java
      ...
1217. public int Precision = 8;
```

[Table of Contents] [Description]

Path 56:

Query Name - **Magic_Numbers**

Severity - **Information**

```
1207. class JpegInfo //jpegencoder.java
      ...
1218. public int NumberOfComponents = 3;
```

[Table of Contents] [Description]

Path 57:

Query Name - **Magic_Numbers**

Severity - **Information**

```
1207. class JpegInfo //jpegencoder.java
        ...
1229. public int Se = 63;
```

[Table of Contents] [Description]

Path 58:

Query Name - **Magic_Numbers**

Severity - **Information**

```
50. private String escapeAndQuote(String value) //csvutil.java
      51. int count = 2;
```

[Table of Contents] [Description]

Path 59:

Query Name - **Magic_Numbers**

Severity - **Information**

```
343. public static BufferedImage getProductionImage(String productVersion, String
productRealeaseDate) //imageutil.java
        ...
346. int IMAGE_WIDTH = 250;
```

[Table of Contents] [Description]

Path 60:

Query Name - **Magic_Numbers**

Severity - **Information**

```
343. public static BufferedImage getProductionImage(String productVersion, String
productRealeaseDate) //imageutil.java
        ...
347. int IMAGE_HEIGHT = 30;
```

[Table of Contents] [Description]

Path 61:

Query Name - **Magic_Numbers**

Severity - **Information**

```
209. public static void sendMail(Collection mailStructCollection) //mailutil.java
        ...
221. int port = 25;
```

[Table of Contents] [Description]

Confusing naming

CWE ID 10032

Description	Naming a method and field with the same name, may be confusing to developers
Likelihood of Exploit	Low
Common Consequences	Difficulty to maintain the code
Potential Mitigations	Renaming methods and fields
Applicable Platforms	All

Path 1:

Query Name - **Confusing_Naming**

Severity - **Information**

```
57. private static class EventInterceptor implements
ValidationEventHandler //validatorimpl.java
      |
      ...
      62. private boolean hadError = false;
```

[Table of Contents] [Description]

Path 2:

Query Name - **Confusing_Naming**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      |
      ...
      1151. private static int maxHotTopics = 10;
```

[Table of Contents] [Description]

Path 3:

Query Name - **Confusing_Naming**

Severity - **Information**

```
44. public class CategoryTreeEvent extends EventObject //categorytreeevent.java
      |
      ...
      52. private boolean hasForum;
```

[Table of Contents] [Description]

Path 4:

Query Name - **Confusing_Naming**

Severity - **Information**

```
44. public class CategoryTreeEvent extends EventObject //categorytreeevent.java
      |
      ...
      54. private boolean hasSubCategory;
```

[Table of Contents] [Description]

Path 5:

Query Name - **Confusing_Naming**

Severity - **Information**

```
21. public class
RebuildAlbumItemIndexTaskMapping //rebuildalbumitemindextaskmapping.java
      ...
      23. private static boolean isRebuilding = false;
```

[Table of Contents] [Description]

Path 6:

Query Name - **Confusing_Naming**

Severity - **Information**

```
62. public class RebuildAttachmentIndexTask extends
TimerTask //rebuildattachmentindextask.java
      ...
      72. private static boolean isRebuilding = false;
```

[Table of Contents] [Description]

Path 7:

Query Name - **Confusing_Naming**

Severity - **Information**

```
61. public class RebuildMemberIndexTask extends
TimerTask //rebuildmemberindextask.java
      ...
      71. private static boolean isRebuilding = false;
```

[Table of Contents] [Description]

Path 8:

Query Name - **Confusing_Naming**

Severity - **Information**

```
61. public class RebuildPostIndexTask extends TimerTask //rebuildpostindextask.java
      ...
      71. private static boolean isRebuilding = false;
```

[Table of Contents] [Description]

Path 9:

Query Name - **Confusing_Naming**

Severity - **Information**

```
390. public static class MailConfig //mvncoreconfig.java
      ...
      394. boolean useMailSource;
```

[Table of Contents] [Description]

Path 10:

Query Name - **Confusing_Naming**

Severity - **Information**

```
390. public static class MailConfig //mvncoreconfig.java
       ...
395. boolean useEmbededSMTPMailServer;
```

[Table of Contents] [Description]

Path 11:

Query Name - **Confusing_Naming**

Severity - **Information**

```
389. public class KeyIterator implements Iterator, Cloneable //configurationkey.java
       ...
404. private boolean hasIndex;
```

[Table of Contents] [Description]

Path 12:

Query Name - **Confusing_Naming**

Severity - **Information**

```
37. public class EnvironmentServiceImplDefault implements
EnvironmentService //environmentserviceimpldefault.java
       ...
39. private boolean isEnterprise;
```

[Table of Contents] [Description]

Path 13:

Query Name - **Confusing_Naming**

Severity - **Information**

```
37. public class EnvironmentServiceImplDefault implements
EnvironmentService //environmentserviceimpldefault.java
       ...
40. private boolean isPortlet;
```

[Table of Contents] [Description]

Path 14:

Query Name - **Confusing_Naming**

Severity - **Information**

```
37. public class EnvironmentServiceImplDefault implements
EnvironmentService //environmentserviceimpldefault.java
       ...
41. private String customizeFor;
```

[Table of Contents] [Description]

Path 15:

Query Name - **Confusing_Naming**

Severity - **Information**

```
85. public class CompressionResponseStream extends
ServletOutputStream //compressionresponsestream.java
      ...
140. protected boolean closed = false;
```

[Table of Contents] [Description]

Path 16:

Query Name - **Confusing_Naming**

Severity - **Information**

```
65. public class DiskFileItem //diskfileitem.java
      ...
111. private boolean isFormField;
```

[Table of Contents] [Description]

Leftover Debug Code

CWE ID	489
Description	Debug code can create unintended entry points in an application.
Common Consequences	The severity of the exposed debug application will depend on the particular instance. At the least, it will give an attacker sensitive information about the settings and mechanics of web applications on the server. At worst, as is often the case, the debug application will allow an attacker complete control over the web application and server, as well as confidential information that either of these access.
Demonstrative Examples	Debug code can be used to bypass authentication. For example, suppose an application has a login script that receives a username and a password. Assume also that a third, optional, parameter, called "debug", is interpreted by the script as requesting a switch to debug mode, and that when this parameter is given the username and password are not checked. In such a case, it is very simple to bypass the authentication process if the special behavior of the application regarding the debug parameter is known. In a case where the form is: <FORM ACTION="/authenticate_login.cgi"> <INPUT TYPE=TEXT name=username> <INPUT TYPE=PASSWORD name=password>

<INPUT TYPE=SUBMIT>
</FORM>

Then a conforming link will look like:

http://TARGET/authenticate_login.cgi?
username=...&password=...

An attacker can change this to:

http://TARGET/authenticate_login.cgi?
username=&password=&debug=1

Which will grant the attacker access to the site, bypassing the authentication process.

Context Notes

A common development practice is to add "back door" code specifically designed for debugging or testing purposes that is not intended to be shipped or deployed with the application. In web-based applications, debug code is used to test and modify web application properties, configuration information, and functions. If a debug application is left on a production server, an attacker may be able to use it to perform these tasks. When this sort of debug code is left in the application, the application is open to unintended modes of interaction. These back door entry points create security risks because they are not considered during design or testing and fall outside of the expected operating conditions of the application.

While it is possible to leave debug code in an application in any language, in J2EE a main method may be a good indicator that debug code has been left in the application, although there may not be any direct security impact.

Node Relationships

ChildOf - Encapsulation (485)
ChildOf - Weaknesses Examined by SAMATE (630)

Source Taxonomies

7 Pernicious Kingdoms - Leftover Debug Code

Applicable Platforms

All

Path 1:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
5. public static void main(String[] args) throws
IOException //filesearchreplace.java
```

[Table of Contents] [Description]

Path 2:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
200. public static void main(String[] args) //phpbbtomvnforum.java
     '
```

[Table of Contents] [Description]

Path 3:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
145. public static void main(String[] args) throws Exception //runningversion.java
     '
```

[Table of Contents] [Description]

Path 4:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
162. public static void main(String[] args) //md5.java
     '
```

[Table of Contents] [Description]

Path 5:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
299. public static void main(String[] args) //utils.java
     '
```

[Table of Contents] [Description]

Path 6:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
1051. public static void main(String[] args) //importexportxml.java
      '
```

[Table of Contents] [Description]

Path 7:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
106. public static void main(String[] args) //md5.java
```

```
    '
```

[Table of Contents] [Description]

Path 8:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
206. public static void main(String[] args) throws Exception //phpbbtomvnforum.java
    '
```

[Table of Contents] [Description]

Path 9:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
285. public static void main(String []args) //utils.java
    '
```

[Table of Contents] [Description]

Path 10:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
311. public static void main(String[] args) throws Exception //converturl.java
    '
```

[Table of Contents] [Description]

Path 11:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
59. public static void main (String args[]) //randomgenerator.java
    '
```

[Table of Contents] [Description]

Path 12:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
217. public static void main (String[] agrs) //dateutil.java
    '
```

[Table of Contents] [Description]

Path 13:

Query Name - **Leftover_Debug_Code**

Severity - **Information**

```
377. public static void main(String[] args) throws Exception //stringutil.java
```

[Table of Contents] [Description]

Missing Catch Block

CWE ID	600
Description	If a Servlet fails to catch all exceptions, it may reveal debugging information that will help an adversary form a plan of attack.
Demonstrative Examples	In the following method a DNS lookup failure will cause the Servlet to throw an exception.

```
protected void doPost (HttpServletRequest req,
HttpServletResponse res)
throws IOException {
  String ip = req.getRemoteAddr();
  InetAddress addr = InetAddress.getByName(ip);
  ...
  out.println("hello " + addr.getHostName());
}
```

Context Notes	When a Servlet throws an exception, the default error response the Servlet container sends back to the user typically includes debugging information. This information is of great value to an attacker. For example, a stack trace might show the attacker a malformed SQL query string, the type of database being used, and the version of the application container. This information enables the attacker to target known vulnerabilities in these components.
Node Relationships	ChildOf - Error Handling (388) ResultsIn - Error Message Information Leaks (209) Peer - Improper Error Handling (390)

Path 1:

Query Name - **Missing_Catch_Block**

Severity - **Information**

```
94. public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException //forumadminservlet.java
```

[Table of Contents] [Description]

Path 2:

Query Name - **Missing_Catch_Block**

Severity - **Information**

```
99. public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException //forumadminservlet.java
     '
```

[Table of Contents] [Description]

Path 3:

Query Name - **Missing_Catch_Block**

Severity - **Information**

```
92. public void doGet(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException //forumuserservlet.java
     '
```

[Table of Contents] [Description]

Path 4:

Query Name - **Missing_Catch_Block**

Severity - **Information**

```
97. public void doPost(HttpServletRequest request, HttpServletResponse response)
throws ServletException, IOException //forumuserservlet.java
     '
```

[Table of Contents] [Description]

Path 5:

Query Name - **Missing_Catch_Block**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
     '
```

[Table of Contents] [Description]

Missing XML Validation

CWE ID	112
Description	Failure to enable validation when parsing XML gives an attacker the opportunity to supply malicious input.
Weakness Ordinality	Primary (Weakness exists independent of other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Potential Mitigations	Always validate XML input against a known XML Schema or DTD. Assume all input is malicious. Use an appropriate combination of black lists and white lists to ensure only valid and expected input is processed by the system.
Context Notes	Most successful attacks begin with a violation of the

programmer's assumptions. By accepting an XML document without validating it against a DTD or XML schema, the programmer leaves a door open for attackers to provide unexpected, unreasonable, or malicious input. It is not possible for an XML parser to validate all aspects of a document's content; a parser cannot understand the complete semantics of the data. However, a parser can do a complete and thorough job of checking the document's structure and therefore guarantee to the code that processes the document that the content is well-formed.

Node Relationships ChildOf - Insufficient Input Validation (20)
Source Taxonomies 7 Pernicious Kingdoms - Missing XML Validation
Applicable Platforms All

Path 1:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
83. public void marshal(Object obj, Result result) throws
JAXBException //marshallerimpl.java
      ...
101. DocumentBuilderFactory dbf = DocumentBuilderFactory.newInstance();
```

[Table of Contents] [Description]

Path 2:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
115. public void updateConfigStepOne(GenericRequest
request) //configurationwebhandler.java
      ...
320. SAXReader reader = new SAXReader();
```

[Table of Contents] [Description]

Path 3:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
578. public void updateConfigStepTwo(GenericRequest
request) //configurationwebhandler.java
      ...
615. SAXReader reader = new SAXReader();
```

[Table of Contents] [Description]

Path 4:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
681. public void updateConfigStepThree(GenericRequest
request) //configurationwebhandler.java
      ...
711. SAXReader reader = new SAXReader();
```

[Table of Contents] [Description]

Path 5:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
767. public void updateConfigMVNCore(GenericRequest
request) //configurationwebhandler.java
      ...
875. SAXReader reader = new SAXReader();
```

[Table of Contents] [Description]

Path 6:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
1103. public void updateUrlPattern(GenericRequest
request) //configurationwebhandler.java
      ...
1117. SAXReader reader = new SAXReader();
```

[Table of Contents] [Description]

Path 7:

Query Name - **Missing_XML_Validation**

Severity - **Information**

```
180. public Document getDocument(String prefix, String
rootName) //configurationxmldocument.java
      ...
189. SAXReader reader = new SAXReader(xmlReader);
```

[Table of Contents] [Description]

Overly-Broad Catch Block

CWE ID	396
Description	Catching overly broad exceptions promotes complex error handling code that is more likely to contain security vulnerabilities.
Demonstrative Examples	The following code excerpt handles three types of exceptions in an identical fashion.

```
try { doExchange(); }
  catch (IOException e) {
    logger.error("doExchange failed", e);
  }
  catch (InvocationTargetException e) {
    logger.error("doExchange failed", e);
  }
  catch (SQLException e) {
    logger.error("doExchange failed", e);
}
```

At first blush, it may seem preferable to deal with these exceptions in a single catch block, as follows: try { doExchange(); } catch (Exception e) { logger.error ("doExchange failed", e); } However, if doExchange() is modified to throw a new type of exception that should be handled in some different kind of way, the broad catch block will prevent the compiler from pointing out the situation. Further, the new catch block will now also handle exceptions derived from RuntimeException such as ClassCastException, and NullPointerException, which is not the programmer's intent.

Context Notes Multiple catch blocks can get ugly and repetitive, but "condensing" catch blocks by catching a high-level class like Exception can obscure exceptions that deserve special treatment or that should not be caught at this point in the program. Catching an overly broad exception essentially defeats the purpose of Java's typed exceptions, and can become particularly dangerous if the program grows and begins to throw new types of exceptions. The new exception types will not receive any attention.

Node Relationships ChildOf - Error Conditions, Return Values, Status Codes (389)

Source Taxonomies 7 Pernicious Kingdoms - Overly-Broad Catch Block

Applicable Platforms C

C++

Java

.NET

Path 1:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
46. public String getMD5_Base64(String input) //mvntomcatjdbcrealm.java
      |
      ...
      |
53. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 2:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
128. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
     ...
141. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 3:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
200. public static void main(String[] args) //phpbbtomvnforum.java
     ...
225. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 4:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
66. public RunningVersion(String phphost, String phpdb, String phpuser, String
phppass, String mvnhost, String mvndb, //runningversion.java
     ...
83. } catch (Exception E) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
125. public static synchronized String getMD5_Base64(String input) //md5.java
     ...
133. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 6:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
132. public Object newInstance( Class
javaContentInterface ) //defaultjaxbcontextimpl.java
     ...
146. } catch( Exception e ) {
```

[Table of Contents] [Description]

Path 7:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
130. public final com.sun.msv.grammar.Grammar getGrammar() throws
JAXBException //grammarinfoimpl.java
      ...
151. } catch( Exception e ) {
```

[Table of Contents] [Description]

Path 8:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
159. public static synchronized String getMD5_Base64(String input) //md5.java
      ...
167. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 9:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
117. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
      ...
137. catch (Exception E)
```

[Table of Contents] [Description]

Path 10:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
151. public static String HauDateTimeFromS(long s, int timeZone) //utils.java
      ...
170. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 11:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
249. static String convertLine(String line) throws Exception //converturl.java
      ...
255. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 12:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
311. public static void main(String[] args) throws Exception //converturl.java
      |
      ...
      |
367. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 13:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
514. private void translate() throws BuildException //translate.java
      |
      ...
      |
534. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 14:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
514. private void translate() throws BuildException //translatetojstl.java
      |
      ...
      |
534. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 15:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
71. public static synchronized OnlineUserFactory getOnlineUserFactory
() //managerfactory.java
      |
      ...
      |
78. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 16:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
86. public static synchronized Authenticator getAuthenticator
() //managerfactory.java
      |
      ...
      |
103. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 17:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
109. public static synchronized RequestProcessor getRequestProcessor
() //managerfactory.java
      ...
115. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 18:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
61. public final class MVNForumConfig //mvnforumconfig.java
      ...
1648. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 19:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1693. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 20:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1701. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 21:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1659. public static void reload() //mvnforumconfig.java
      ...
1974. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 22:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
56. public class MVNForumFactoryConfig //mvnforumfactoryconfig.java
      ...
132. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 23:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
119. public static String getMemberTitle(int postCount) //myutil.java
      ...
131. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 24:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
50. public final class AdminModuleConfig //adminmoduleconfig.java
      ...
72. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 25:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
77. public void prepareConfigMVNForum(GenericRequest
request) //configurationwebhandler.java
      ...
88. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 26:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
96. public void prepareConfigMVNCore(GenericRequest
request) //configurationwebhandler.java
```

```
...
107. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 27:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
...
182. try { out.close(); } catch (Exception ee) {}
```

[Table of Contents] [Description]

Path 28:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
122. public static void exportZip(String filename, HttpServletRequest
request, //exportwebhelper.java
...
184. try { new File(zipFilePath).delete(); } catch (Exception ee) {}
```

[Table of Contents] [Description]

Path 29:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
410. public static void exportGroupList(XMLWriter xmlWriter) //groupxml.java
...
422. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 30:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
...
272. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 31:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
131. public static void importXmlZip(HttpServletRequest request, HttpServletResponse
response) //importwebhandler.java
      ...
283. } catch (Exception e) {}
```

[Table of Contents] [Description]

Path 32:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
103. public void processAdd(GenericRequest request) //memberwebhandler.java
      ...
228. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 33:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
      ...
993. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 34:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
      ...
995. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 35:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
      ...
998. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 36:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
981. public static void exportMemberList(XMLWriter xmlWriter) //memberxml.java
       ...
1000. catch (Exception e) { /* doesn't exist => ignore */ }
```

[Table of Contents] [Description]

Path 37:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
125. public static String convertPassword(String jiveEncodedPassword) //jivexml.java
       ...
130. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 38:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
206. public static void checkRegisteredMembersGroup() throws
CreateException, //mvnforumxml.java
       ...
214. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 39:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
125. boolean hasPermssionInAtLeastOneForum() //forumlistpermission.java
       ...
147. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 40:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
179. public void reloadPermission() //onlineuserimpl.java
       ...
188. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 41:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
193. public void reloadProfile() //onlineuserimpl.java
        ...
210. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 42:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
215. public boolean updateNewMessageCount(boolean forceUpdate) //onlineuserimpl.java
        ...
234. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 43:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
        ...
199. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 44:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
248. protected OnlineUser login(GenericRequest request, GenericResponse
response, //onlineusermanager.java
        ...
254. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
        ...
309. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 46:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
      |...
      |91. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 47:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
57. public static synchronized MvnAuthService getMvnAuthService
() //mvnauthservicefactory.java
      |...
      |64. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 48:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
258. public String drawCategory(CategoryTreeEvent event) //categoryselector.java
      |...
      |268. } catch (Exception ex) {}
```

[Table of Contents] [Description]

Path 49:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
288. public String drawCategory(CategoryTreeEvent event) //listforums.java
      |...
      |314. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
131. public static void deleteAttachments_inForum(int forumID) throws
DatabaseException //attachmentutil.java
      |
```

```
...
156. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 51:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
127. public void run() //deleteorphanpmattachmenttask.java
...
133. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 52:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
92. public void run() //reloadtomcattimertaskext.java
...
105. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 53:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
67. public static void createAccount(String memberName, GenericRequest
request) //remoteuserutil.java
...
123. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 54:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
82. public static void addNewMember(HttpServletRequest request, String memberName,
String memberPassword, String memberEmail, Date memberBirthday) //kgutils.java
...
170. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 55:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
58. public class DAOFactory //daofactory.java
      ...
237. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 56:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
70. public void run() //addupdateattachmentindextask.java
      ...
83. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 57:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
116. private static void initializeAnalyzer() //attachmentindexer.java
      ...
128. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 58:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
61. public void run() //deleteattachmentindextask.java
      ...
65. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 59:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
70. public void run() //addupdatememberindextask.java
      ...
83. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 60:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
62. public void run() //deletememberindextask.java
      |...
      66. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 61:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
117. private static void initializeAnalyzer() //memberindexer.java
      |...
      129. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 62:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
70. public void run() //addupdatepostindextask.java
      |...
      83. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 63:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
69. public void run() //deletepostindextask.java
      |...
      85. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 64:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
127. private static void initializeAnalyzer() //postindexer.java
      |...
      139. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 65:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
      ...
83. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 66:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
76. public void run() //updatethreadindextask.java
      ...
115. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 67:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
57. public static MvnForumService getMvnForumService() //mvnforumservicefactory.java
      ...
63. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 68:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
      ...
316. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 69:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
      ...
333. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 70:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
300. public void deleteData(String category, String nameId, BinaryStorageHandle
handle) //binarystorageserviceimpldisk.java
      ...
358. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 71:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
146. public void processAdd(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
      ...
296. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 72:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
729. static void deleteAttachments_inThread(int threadID) throws
DatabaseException //attachmentwebhandler.java
      ...
749. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 73:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
129. public void processAdd(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
351. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 74:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
70. public void prepareAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
95. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 75:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
124. public void processAdd(GenericRequest request, GenericResponse
response) //pmattachmentwebhandler.java
      ...
288. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 76:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
506. public void deleteOrphanPmAttachment() throws
DatabaseException //pmattachmentwebhandler.java
      ...
535. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 77:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
90. public void prepareAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
115. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 78:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1418. public void prepareListRSS(GenericRequest request) //threadwebhandler.java
      ...
1435. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 79:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
50. public class UserModuleConfig //usermoduleconfig.java
```

```
...
71. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 80:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1. <%-- //changememberstatussuccess.jsp
...
52. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 81:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1. <%-- //configindex.jsp
...
121. } catch (Exception e) {}
```

[Table of Contents] [Description]

Path 82:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1. <%-- //listattachments.jsp
...
148. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 83:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1. <%-- //printthread.jsp
...
92. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 84:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1. <%-- //viewthread.jsp
```

```
...
325. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 85:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
292. public static void reload() //mvncoreconfig.java
...
384. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 86:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
44. public class SecondaryMVNCoreConfig //secondarymvncoreconfig.java
...
63. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 87:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
71. private DBConnectionManager() //dbconnectionmanager.java
...
75. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 88:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
88. private DBConnectionManager(DBOptions dbOptions) //dbconnectionmanager.java
...
92. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 89:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
134. public static int getDatabaseType() //dbutils.java
```

```
...
168. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 90:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
238. public static Connection getConnection() throws SQLException //dbutils.java
...
251. } catch (Exception ex) { }
```

[Table of Contents] [Description]

Path 91:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
120. public static int getDatabaseType() //dbutils2.java
...
147. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 92:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
162. public static Connection getConnection() throws SQLException //dbutils2.java
...
175. } catch (Exception ex) { }
```

[Table of Contents] [Description]

Path 93:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
81. public String getString(Locale locale, String key) //cacheresourcebundle.java
...
85. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 94:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
90. public String getString(Locale locale, String key, Object[]
```

```
args) //cacheresourcebundle.java
      ...
      101. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 95:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
56. public DatabaseInfo() //databaseinfo.java
      ...
      67. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 96:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
69. public static String getProperty(String key) //systeminfo.java
      ...
      73. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 97:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
78. private void loadContentInterceptor(String
contentInterceptorClassName) //interceptorservice.java
      ...
      86. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 98:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
91. private void loadMailInterceptor(String
mailInterceptorClassName) //interceptorservice.java
      ...
      99. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 99:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
104. private void loadLoginIDInterceptor(String
loginIDInterceptorClassName) //interceptorservice.java
      ...
112. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 100:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
117. private void loadPasswordInterceptor(String
passwordInterceptorClassName) //interceptorservice.java
      ...
126. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 101:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
70. public static synchronized String getMD5_Base64(String input) //encoder.java
      ...
78. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 102:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
114. public static String decrypt_AES_from_HEX(String input, String
key) //encoder.java
      ...
122. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 103:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
174. public static String filterUrl(String url) //encoder.java
      ...
202. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 104:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
50. public static synchronized MvnCoreService getMvnCoreService
() //mvncoreservicefactory.java
    ...
58. } catch (Exception e) {
```

[Table of Contents] [Description]

Path 105:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
46. public final class IPFilterServiceImplDefault implements
IPFilterService //ipfilterserviceimpldefault.java
    ...
71. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 106:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
1264. private void getYCCArray() //jpegencoder.java
    ...
1304. catch (Exception e)
```

[Table of Contents] [Description]

Path 107:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
300. public static void writeJpegImage_nonSun(Image image, OutputStream
outputStream) throws IOException //imageutil.java
    ...
307. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 108:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
102. public static void checkGoodEmail(String input) throws
BadInputException //mailutil.java
    ...
140. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 109:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
473. public static MailMessageStruct[] receiveEmail(String popServer, String
popUser, String popPassword) //mailutil.java
      ...
514. } catch (Exception ex) {
```

[Table of Contents] [Description]

Path 110:

Query Name - **Overly_Broad_Catch**

Severity - **Information**

```
389. public static int parseIntSizeValue(String propertyValue, int
defaultValue) //stringutil.java
      ...
407. } catch (Exception e) {
```

[Table of Contents] [Description]

Overly-Broad Throws Declaration

CWE ID 397

Description Throwing overly broad exceptions promotes complex error handling code that is more likely to contain security vulnerabilities.

Demonstrative Examples The following method throws three types of exceptions.

```
public void doExchange() throws IOException,
InvocationTargetException, SQLException {
  ...
}
```

Context Notes While it might seem tidier to write public void doExchange() throws Exception { ... } doing so hampers the caller's ability to understand and handle the exceptions that occur. Further, if a later revision of doExchange() introduces a new type of exception that should be treated differently than previous exceptions, there is no easy way to enforce this requirement. Declaring a method to throw Exception or Throwable makes it difficult for callers to do good error handling and error recovery. Java's exception mechanism is set up to make it easy for callers to anticipate what can go wrong and write code to handle each specific exceptional circumstance. Declaring that a method throws a generic form of exception defeats this system.

Node Relationships ChildOf - Error Conditions, Return Values, Status Codes (389)
Source Taxonomies 7 Pernicious Kingdoms - Overly-Broad Throws Declaration
Applicable Platforms C

C++

Java

.NET

Path 1:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
145. public static void main(String[] args) throws Exception //runningversion.java
     '
```

[Table of Contents] [Description]

Path 2:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
206. public static void main(String[] args) throws Exception //phpbbtomvnforum.java
     '
```

[Table of Contents] [Description]

Path 3:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
249. static String convertLine(String line) throws Exception //converturl.java
     '
```

[Table of Contents] [Description]

Path 4:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
287. public static void convertFile(String srcFilename, String
descFilename) //converturl.java
     '
```

[Table of Contents] [Description]

Path 5:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
311. public static void main(String[] args) throws Exception //converturl.java
     |
```

[Table of Contents] [Description]

Path 6:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
97. public void begin(String namespace, String name, Attributes
attributes) //setparentrule.java
     |
```

[Table of Contents] [Description]

Path 7:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
102. public void begin(Attributes attributes) throws Exception //setparentrule.java
     |
```

[Table of Contents] [Description]

Path 8:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
152. public Configuration getConfiguration() throws
Exception //configurationfactory.java
     |
```

[Table of Contents] [Description]

Path 9:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
451. public Object createObject(Attributes attribs) throws
Exception //configurationfactory.java
     |
```

[Table of Contents] [Description]

Path 10:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
483. public Object createObject(Attributes attributes) throws
Exception //configurationfactory.java
```

```
    '
```

[Table of Contents] [Description]

Path 11:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
153. public void load() throws Exception //dom4jconfiguration.java
    '
```

[Table of Contents] [Description]

Path 12:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
145. public void load() throws Exception //hierarchicaldom4jconfiguration.java
    '
```

[Table of Contents] [Description]

Path 13:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
155. public void load(URL url) throws
Exception //hierarchicaldom4jconfiguration.java
    '
```

[Table of Contents] [Description]

Path 14:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
377. public static void main(String[] args) throws Exception //stringutil.java
    '
```

[Table of Contents] [Description]

Path 15:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
164. void write(File file) throws Exception; //fileitem.java
    '
```

[Table of Contents] [Description]

Path 16:

Query Name - **Overly_Broad_Throws**

Severity - **Information**

```
386. public void write(File file) throws Exception //diskfileitem.java
```

[Table of Contents] [Description]

	Use of system output stream
CWE ID	10033
Description	A dedicated logging class should be used, rather than using default output stream, to ease the monitoring process
Likelihood of Exploit	Low
Common Consequences	Difficulty to maintain the application and to follow log files
Potential Mitigations	Using dedicated logging class
Applicable Platforms	All

Path 1:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
5. public static void main(String[] args) throws
IOException //filesearchreplace.java
    ...
    9. System.out.println("Utility to update file contents using simple
    search/replace strings");
```

[Table of Contents] [Description]

Path 2:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
198. public final static void waitln() //utility.java
    199. System.out.println("\n..press <return> to continue");
```

[Table of Contents] [Description]

Path 3:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
71. protected String digest(String credentials) //mvntomcatjdbcrealm.java
    72. System.out.println("MVNTomcatJDBCRealm.digest()");
```

[Table of Contents] [Description]

Path 4:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
158. public static void migrateUsers(PrintStream ostream, int
gender) //migrator.java
        ...
289. System.out.println(userCount);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
128. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
        ...
142. System.err.println("Unable to load driver.");
```

[Table of Contents] [Description]

Path 6:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
160. public void convert() //phpbbtomvnforum.java
        ...
162. System.out.println("Conversion started...");
```

[Table of Contents] [Description]

Path 7:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
66. public RunningVersion(String phphost, String phpdb, String phpuser, String
phppass, String mvnhost, String mvndb, //runningversion.java
        ...
85. E.printStackTrace();
```

[Table of Contents] [Description]

Path 8:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
102. public void convert() //runningversion.java
        ...
105. System.out.println("Conversion started...");
```

[Table of Contents] [Description]

Path 9:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
162. public static void main(String[] args) //md5.java
       ...
190. System.out.println("PHPBB : " + phpbbPassword);
```

[Table of Contents] [Description]

Path 10:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
       ...
96. System.out.println("Error: Does not support GENDER = " + GENDER);
```

[Table of Contents] [Description]

Path 11:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
299. public static void main(String[] args) //utils.java
       ...
302. System.out.println(Utils.stripPHPBBQuotes(phpstr));
```

[Table of Contents] [Description]

Path 12:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
853. public void importMember() //importexportxml.java
       ...
869. System.out.println("memberName " + memberName);
```

[Table of Contents] [Description]

Path 13:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
81. public List importGroupTypes() throws JAXBException //grouplistdao.java
       ...
85. System.out.println(groupListType);
```

[Table of Contents] [Description]

Path 14:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
132. protected final void dump() //abstractunmarshallingeventhandlerimpl.java
     |133. System.err.println("state is :"+state);
```

[Table of Contents] [Description]

Path 15:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
106. public static void main(String[] args) //md5.java
     |...
     |126. System.out.print("PHPBB :");
```

[Table of Contents] [Description]

Path 16:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
106. public static void main(String[] args) //md5.java
     |...
     |133. System.out.println(printable);
```

[Table of Contents] [Description]

Path 17:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
     |...
     |283. System.out.println(userCount + ":" + MemberName + " updated!");
```

[Table of Contents] [Description]

Path 18:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
117. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
     |...
     |139. System.err.println("Unable to load driver.");
```

[Table of Contents] [Description]

Path 19:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
     ...
162. System.out.println("Conversion started...");
```

[Table of Contents] [Description]

Path 20:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
285. public static void main(String []args) //utils.java
     ...
318. System.out.println(DateTimeFromS(1146518792));
```

[Table of Contents] [Description]

Path 21:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
85. static String convertTagA(String tagA) throws
IllegalArgumentException //converturl.java
     ...
99. System.out.println("WARNING: TagA does not have href: " + tagA);
```

[Table of Contents] [Description]

Path 22:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
297. public static void ExportContentToFile(ArrayList lines, String
fileName) //converturl.java
     ...
307. System.out.print(e);
```

[Table of Contents] [Description]

Path 23:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
996. public Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order) throws IllegalArgumentException,
DatabaseException //forumdaoimpljdbc.java
```

```
...
1037. System.out.println("===sql mysql: " + sql);
```

[Table of Contents] [Description]

Path 24:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
69. public synchronized void addMvnForumEventListener(MvnForumEventListener
listener) //mvnforumeventmanager.java
    70. System.out.println("MvnForumEventManager.addMvnForumEventListener()");
```

[Table of Contents] [Description]

Path 25:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
755. public void prepareListAttachments(GenericRequest request, GenericResponse
response) //attachmentwebhandler.java
    ...
    787. System.out.println("totalAttachments : " + totalAttachments);
```

[Table of Contents] [Description]

Path 26:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addcategory.jsp
    ...
    131. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 27:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addforum.jsp
    ...
    123. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 28:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //assignforumtogroup.jsp
       ...
     144. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 29:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //assignforumtomember.jsp
       ...
     140. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 30:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //editforum.jsp
       ...
     132. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 31:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //forummanagement.jsp
       ...
     235. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 32:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //groupmanagement.jsp
       ...
     117. out.print(groupBean.getGroupOwnerName());
```

[Table of Contents] [Description]

Path 33:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //permissionsummary.jsp
       ...
219. out.print(groupBean.getGroupOwnerName());
```

[Table of Contents] [Description]

Path 34:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //permissionsummary.jsp
       ...
268. out.print(groupBean.getGroupOwnerName());
```

[Table of Contents] [Description]

Path 35:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //viewgroup.jsp
       ...
124. out.print(objGroupsBean.getGroupOwnerName());
```

[Table of Contents] [Description]

Path 36:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addmessage.jsp
       ...
343. out.print(EnableEmotionFilter.filter(messageIcon + "  ",
contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 37:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addpost.jsp
       ...
393. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 38:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addpost.jsp
      |
      ...
      442. out.print(EnableEmotionFilter.filter(prePostIcon + "  ",
      contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 39:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //addwatch.jsp
      |
      ...
      175. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 40:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //index.jsp
      |
      ...
      127. out.println(request.getAttribute("tree"));
```

[Table of Contents] [Description]

Path 41:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //index.jsp
      |
      ...
      418. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 42:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listattachments.jsp
      |
      ...
      179. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 43:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listforums.jsp
       ...
      97. out.println(request.getAttribute("tree"));
```

[Table of Contents] [Description]

Path 44:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listforums.jsp
       ...
     462. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 45:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listonlineusers.jsp
       ...
     160. out.print(desc);
```

[Table of Contents] [Description]

Path 46:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listrecentthreads.jsp
       ...
     125. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 47:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listthreads.jsp
       ...
     115. out.println(request.getAttribute("tree"));
```

[Table of Contents] [Description]

Path 48:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listthreads.jsp
       ...
148. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 49:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //listunansweredthreads.jsp
       ...
148. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 50:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //modcp.jsp
       ...
165. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 51:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //moderatependingposts.jsp
       ...
217. out.print(EnableEmotionFilter.filter(postBean.getPostIcon() +
"  ", contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 52:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //moderatependingthreads.jsp
       ...
253. out.print(EnableEmotionFilter.filter(postBean.getPostIcon() +
"  ", contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 53:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //movethread.jsp
      ...
182. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 54:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //printpost.jsp
      ...
121. out.print(EnableEmotionFilter.filter(postBean.getPostIcon() +
"  ", contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 55:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //printthread.jsp
      ...
169. out.print(EnableEmotionFilter.filter(postBean.getPostIcon() +
"  ", contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 56:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //rsssummary.jsp
      ...
178. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 57:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //search.jsp
      ...
201. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 58:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //viewmessage.jsp
      ...
      207. out.print(EnableEmotionFilter.filter(messageBean.getMessageIcon() +
      "  ", contextPath + MVNForumGlobal.EMOTION_DIR));
```

[Table of Contents] [Description]

Path 59:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //viewthread.jsp
      ...
      343. out.println(request.getAttribute("Result"));
```

[Table of Contents] [Description]

Path 60:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
1. <%-- //viewthread.jsp
      ...
      862. out.print(name);
```

[Table of Contents] [Description]

Path 61:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
89. public static void dump(Configuration configuration, PrintStream
out) //configurationutils.java
      ...
      95. out.print(key);
```

[Table of Contents] [Description]

Path 62:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
89. public static void dump(Configuration configuration, PrintStream
out) //configurationutils.java
      ...
      96. out.print("=");
```

[Table of Contents] [Description]

Path 63:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
89. public static void dump(Configuration configuration, PrintStream
out) //configurationutils.java
      |...
      97. out.print(value);
```

[Table of Contents] [Description]

Path 64:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
89. public static void dump(Configuration configuration, PrintStream
out) //configurationutils.java
      |...
      100. out.println();
```

[Table of Contents] [Description]

Path 65:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      |...
      117. out.print(key);
```

[Table of Contents] [Description]

Path 66:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      |...
      118. out.print("=");
```

[Table of Contents] [Description]

Path 67:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
```

```
out) //configurationutils.java
      ...
      119. out.print(value);
```

[Table of Contents] [Description]

Path 68:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
111. public static void dump(Configuration configuration, PrintWriter
out) //configurationutils.java
      ...
      123. out.println();
```

[Table of Contents] [Description]

Path 69:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
59. public static void main (String args[]) //randomgenerator.java
      60. System.out.println("getRandomString = " + getRandomString());
```

[Table of Contents] [Description]

Path 70:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      53. out.println("<html>");
```

[Table of Contents] [Description]

Path 71:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      54. out.println("<head>");
```

[Table of Contents] [Description]

Path 72:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      55. out.println("<title>Hello World Servlet</title>");
```

[Table of Contents] [Description]

Path 73:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      56. out.println("</head>");
```

[Table of Contents] [Description]

Path 74:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      57. out.println("<body>");
```

[Table of Contents] [Description]

Path 75:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      58. out.println("<h1>Hello World Servlet run now.</h1>");
```

[Table of Contents] [Description]

Path 76:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
      59. out.println("</body>");
```

[Table of Contents] [Description]

Path 77:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
47. public void doGet(HttpServletRequest request, HttpServletResponse
response) //helloworldservlet.java
      ...
60. out.println("</html>");
```

[Table of Contents] [Description]

Path 78:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
91. public void Compress() //jpegencoder.java
      ...
99. System.out.println("IO Error: " + e.getMessage());
```

[Table of Contents] [Description]

Path 79:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
958. void bufferIt(BufferedOutputStream outStream, int code,int
size) //jpegencoder.java
      ...
976. System.out.println("IO Error: " + e.getMessage());
```

[Table of Contents] [Description]

Path 80:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
217. public static void main (String[] agrs) //dateutil.java
      ...
219. System.out.println(duration);
```

[Table of Contents] [Description]

Path 81:

Query Name - **Use_of_System_Output_Stream**

Severity - **Information**

```
377. public static void main(String[] args) throws Exception //stringutil.java
```

```
...
386. System.out.println("r = [" + StringUtil.getShorterString(s1, 22) + "]");
```

[Table of Contents] [Description]

SQL Injection

CWE ID	89
Description	SQL injection attacks are another instantiation of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.
Likelihood of Exploit	Very High
Common Consequences	Confidentiality: Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL injection vulnerabilities. Authentication: If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password. Authorization: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL injection vulnerability. Integrity: Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL injection attack.
Potential Mitigations	Requirements specification: A non-SQL style database which is not subject to this flaw may be chosen. Design: Follow the principle of least privilege when creating user accounts to a SQL database. Users should only have the minimum privileges necessary to use their account. If the requirements of the system indicate that a user can read and modify their own data, then limit their privileges so they cannot read/write others' data. Design: Duplicate any filtering done on the client-side on the server side. Implementation: Implement SQL strings using prepared statements that bind variables. Prepared statements that do not bind variables can be vulnerable to attack. Implementation: Use vigorous white-list style checking on any user input that may be used in a SQL command. Rather than escape meta-characters, it is safest to disallow them entirely. Reason: Later use of data that have been entered in the database may neglect to escape meta-characters before use. Narrowly define the set of

safe characters based on the expected value of the parameter in the request.

Demonstrative Examples

The following code dynamically constructs and executes a SQL query that searches for items matching a specified name. The query restricts the items displayed to those where owner matches the user name of the currently-authenticated user.

C# Example:

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '"
+ userName + "' AND itemname = '" + ItemName.Text +
"'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
...
```

The query that this code intends to execute follows: SELECT * FROM items WHERE owner = <userName> AND itemname = <itemName>; However, because the query is constructed dynamically by concatenating a constant base query string and a user input string, the query only behaves correctly if itemName does not contain a single-quote character. If an attacker with the user name wiley enters the string "name' OR 'a'='a" for itemName, then the query becomes the following: SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a'; The addition of the OR 'a'='a' condition causes the where clause to always evaluate to true, so the query becomes logically equivalent to the much simpler query: SELECT * FROM items; This simplification of the query allows the attacker to bypass the requirement that the query only return items owned by the authenticated user; the query now returns all entries stored in the items table, regardless of their specified owner.

This example examines the effects of a different malicious value passed to the query constructed and executed in the above example. If an attacker with the user name hacker enters the string "hacker'); DELETE FROM items; --" for itemName, then the query becomes the following two queries:

SQL Example:

```
SELECT * FROM items WHERE owner = 'wiley' AND
itemname = 'name';
DELETE FROM items;
--'
```

Many database servers, including Microsoft(R) SQL Server 2000, allow multiple SQL statements separated by semicolons to be executed at once. While this attack string results in an error on Oracle and other database servers that do not allow the batch-execution of statements separated by semicolons, on databases that do allow batch execution, this type of attack allows the attacker to execute arbitrary commands against the database. Notice the trailing pair of hyphens (--), which specifies to most database servers that the remainder of the statement is to be treated as a comment and not executed [19]. In this case the comment character serves to remove the trailing single-quote left over from the modified query. On a database where comments are not allowed to be used in this way, the general attack could still be made effective using a trick similar to the one shown in Example 1. If an attacker enters the string "name'); DELETE FROM items; SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name'; DELETE FROM items; SELECT * FROM items WHERE 'a'='a'; One traditional approach to preventing SQL injection attacks is to handle them as an input validation problem and either accept only characters from a whitelist of safe values or identify and escape a blacklist of potentially malicious values. Whitelisting can be a very effective means of enforcing strict input validation rules, but parameterized SQL statements require less maintenance and can offer more guarantees with respect to security. As is almost always the case, blacklisting is riddled with loopholes that make it ineffective at preventing SQL injection attacks. For example, attackers can: - Target fields that are not quoted - Find ways to bypass the need for certain escaped meta-characters - Use stored procedures to hide the injected meta-characters Manually escaping characters in input to SQL queries can help, but it will not make your application secure from SQL injection attacks. Another solution commonly proposed for dealing with SQL injection attacks is to use stored procedures. Although stored procedures prevent some types of SQL injection attacks, they fail to protect against many others. For example, the following PL/SQL procedure is vulnerable to the same SQL injection attack shown in the first example. procedure get_item (itm_cv IN OUT ItmCurTyp, usr in varchar2, itm in varchar2) is open itm_cv for ' SELECT * FROM items WHERE ' || 'owner = '|| usr || ' AND itemname = ' || itm || '; end get_item; Stored procedures typically help prevent SQL injection attacks by limiting the types of statements that can be passed to their parameters. However, there are many ways around the limitations and many interesting

statements that can still be passed to stored procedures. Again, stored procedures can prevent some exploits, but they will not make your application secure against SQL injection attacks.

MS SQL has a built in function that enables shell command execution. An SQL injection in such a context could be disastrous. For example, a query of the form:

SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='$user_input' ORDER BY PRICE

Where $user_input is taken from the user and unfiltered.

If the user provides the string:

' exec master..xp_cmdshell 'vol' --

The query will take the following form: "

SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' exec master..xp_cmdshell 'vol' --' ORDER BY PRICE

Now, this query can be broken down into: [1] a first SQL query: SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' [2] a second SQL query, which executes a shell command: exec master..xp_cmdshell 'vol' [3] an MS SQL comment: --' ORDER BY PRICE As can be seen, the malicious input changes the semantics of the query into a query, a shell command execution and a comment.

Observed Examples

CVE-2004-0366 -

CVE-2004-0343 -

CVE-2003-0779 -

CVE-2003-0500 -

CVE-2003-0377 -

Context Notes

SQL injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. Essentially, the attack is accomplished by placing a meta character into data input

to then place SQL commands in the control plane, which did not exist there before. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

If successful, SQL Injection attacks can give an attacker access to backend database contents, the ability to remotely execute system commands, or in some circumstances the means to take control of the Windows server hosting the database.

Dynamically generating queries that include user input can lead to SQL injection attacks. An attacker can insert SQL commands or modifiers in the user input that can cause the query to behave in an unsafe manner.

Constructing a dynamic SQL statement with user input may allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Factors: resultant to special character mismanagement, MAID, or blacklist/whitelist problems. Can be primary to authentication errors.

References M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.

Node Relationships
ChildOf - Injection (74)
ChildOf - Weaknesses in OWASP Top Ten (629)
ChildOf - Weaknesses Examined by SAMATE (630)
ChildOf - Weaknesses Used by NVD (635)
ParentOf - SQL Injection: Hibernate (564)

Source Taxonomies
PLOVER - SQL injection

7 Pernicious Kingdoms - SQL Injection

CLASP - SQL injection

Applicable Platforms All

Path 1:

Query Name - **SQL_injection**

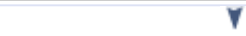
Severity - **Critical**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
      ...
      642. String order = GenericParamUtil.getParameter(request, "order");
      ...
```

```
671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
(offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1468. return getBeans_withSortSupport_limit_mysql(offset, rowsToReturn, sort,
order, memberStatus);
```



```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1541. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
1542. sql.append(" LIMIT ?, ?");
...
1546. statement = connection.prepareStatement(sql.toString());
...
1549. resultSet = statement.executeQuery();
```

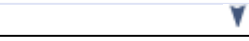
[Table of Contents] [Description]

Path 2:

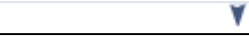
Query Name - **SQL_injection**

Severity - **Critical**

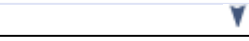
```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
59. String ret = request.getParameter(param);
...
61. return ret.trim();
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
642. String order = GenericParamUtil.getParameter(request, "order");
...
671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
(offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1470. return getBeans_withSortSupport_limit_noscroll(offset, rowsToReturn,
sort, order, memberStatus);
```



```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1672. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
...
1676. statement = connection.prepareStatement(sql.toString());
...
```

```
1678. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 3:

Query Name - **SQL_injection**

Severity - **Critical**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
    59. String ret = request.getParameter(param);
    ...
    61. return ret.trim();
```

▼

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    642. String order = GenericParamUtil.getParameter(request, "order");
    ...
    671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
    (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

▼

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1472. return getBeans_withSortSupport_limit_general(offset, rowsToReturn,
    sort, order, memberStatus);
```

▼

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1807. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
    ...
    1811. statement = connection.prepareStatement(sql.toString(),
    ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
    ...
    1818. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

SQL Injection

CWE ID	89
Description	SQL injection attacks are another instantiation of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.
Likelihood of Exploit	Very High
Common Consequences	Confidentiality: Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL injection vulnerabilities.

Authentication: If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password.

Authorization: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL injection vulnerability.

Integrity: Just as it may be possible to read sensitive information, it is also possible to make changes or even delete this information with a SQL injection attack.

Potential Mitigations

Requirements specification: A non-SQL style database which is not subject to this flaw may be chosen.

Design: Follow the principle of least privilege when creating user accounts to a SQL database. Users should only have the minimum privileges necessary to use their account. If the requirements of the system indicate that a user can read and modify their own data, then limit their privileges so they cannot read/write others' data.

Design: Duplicate any filtering done on the client-side on the server side.

Implementation: Implement SQL strings using prepared statements that bind variables. Prepared statements that do not bind variables can be vulnerable to attack.

Implementation: Use vigorous white-list style checking on any user input that may be used in a SQL command. Rather than escape meta-characters, it is safest to disallow them entirely. Reason: Later use of data that have been entered in the database may neglect to escape meta-characters before use. Narrowly define the set of safe characters based on the expected value of the parameter in the request.

Demonstrative Examples

The following code dynamically constructs and executes a SQL query that searches for items matching a specified name. The query restricts the items displayed to those where owner matches the user name of the currently-authenticated user.

C# Example:

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '"
+ userName + "' AND itemname = '" + ItemName.Text +
"'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
```

```
sda.Fill(dt);
...
```

The query that this code intends to execute follows: SELECT * FROM items WHERE owner = <userName> AND itemname = <itemName>; However, because the query is constructed dynamically by concatenating a constant base query string and a user input string, the query only behaves correctly if itemName does not contain a single-quote character. If an attacker with the user name wiley enters the string "name' OR 'a'='a" for itemName, then the query becomes the following: SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a'; The addition of the OR 'a'='a' condition causes the where clause to always evaluate to true, so the query becomes logically equivalent to the much simpler query: SELECT * FROM items; This simplification of the query allows the attacker to bypass the requirement that the query only return items owned by the authenticated user; the query now returns all entries stored in the items table, regardless of their specified owner.

This example examines the effects of a different malicious value passed to the query constructed and executed in the above example. If an attacker with the user name hacker enters the string "hacker'); DELETE FROM items; --" for itemName, then the query becomes the following two queries:

SQL Example:

```
SELECT * FROM items WHERE owner = 'wiley' AND
itemname = 'name';
DELETE FROM items;
--'
```

Many database servers, including Microsoft(R) SQL Server 2000, allow multiple SQL statements separated by semicolons to be executed at once. While this attack string results in an error on Oracle and other database servers that do not allow the batch-execution of statements separated by semicolons, on databases that do allow batch execution, this type of attack allows the attacker to execute arbitrary commands against the database. Notice the trailing pair of hyphens (--), which specifies to most database servers that the remainder of the statement is to be treated as a comment and not executed [19]. In this case the comment character serves to remove the trailing single-quote left over from the modified query. On a

database where comments are not allowed to be used in this way, the general attack could still be made effective using a trick similar to the one shown in Example 1. If an attacker enters the string "name'); DELETE FROM items; SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name'; DELETE FROM items; SELECT * FROM items WHERE 'a'='a'; One traditional approach to preventing SQL injection attacks is to handle them as an input validation problem and either accept only characters from a whitelist of safe values or identify and escape a blacklist of potentially malicious values. Whitelisting can be a very effective means of enforcing strict input validation rules, but parameterized SQL statements require less maintenance and can offer more guarantees with respect to security. As is almost always the case, blacklisting is riddled with loopholes that make it ineffective at preventing SQL injection attacks. For example, attackers can: - Target fields that are not quoted - Find ways to bypass the need for certain escaped meta-characters - Use stored procedures to hide the injected meta-characters Manually escaping characters in input to SQL queries can help, but it will not make your application secure from SQL injection attacks. Another solution commonly proposed for dealing with SQL injection attacks is to use stored procedures. Although stored procedures prevent some types of SQL injection attacks, they fail to protect against many others. For example, the following PL/SQL procedure is vulnerable to the same SQL injection attack shown in the first example. procedure get_item (itm_cv IN OUT ItmCurTyp, usr in varchar2, itm in varchar2) is open itm_cv for ' SELECT * FROM items WHERE ' || 'owner = '|| usr || ' AND itemname = ' || itm || '; end get_item; Stored procedures typically help prevent SQL injection attacks by limiting the types of statements that can be passed to their parameters. However, there are many ways around the limitations and many interesting statements that can still be passed to stored procedures. Again, stored procedures can prevent some exploits, but they will not make your application secure against SQL injection attacks.

MS SQL has a built in function that enables shell command execution. An SQL injection in such a context could be disastrous. For example, a query of the form:

```
SELECT ITEM,PRICE FROM PRODUCT WHERE
ITEM_CATEGORY='$user_input' ORDER BY PRICE
```

Where $user_input is taken from the user and unfiltered.

If the user provides the string:

' exec master..xp_cmdshell 'vol' --

The query will take the following form: "

SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' exec master..xp_cmdshell 'vol' --' ORDER BY PRICE

Now, this query can be broken down into: [1] a first SQL query: SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' [2] a second SQL query, which executes a shell command: exec master..xp_cmdshell 'vol' [3] an MS SQL comment: --' ORDER BY PRICE As can be seen, the malicious input changes the semantics of the query into a query, a shell command execution and a comment.

Observed Examples

CVE-2004-0366 -

CVE-2004-0343 -

CVE-2003-0779 -

CVE-2003-0500 -

CVE-2003-0377 -

Context Notes

SQL injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. Essentially, the attack is accomplished by placing a meta character into data input to then place SQL commands in the control plane, which did not exist there before. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

If successful, SQL Injection attacks can give an attacker access to backend database contents, the ability to remotely execute system commands, or in some circumstances the means to take control of the Windows server hosting the database.

Dynamically generating queries that include user input can lead to SQL injection attacks. An attacker can insert SQL commands or modifiers in the user input that can cause the query to behave in an unsafe manner.

Constructing a dynamic SQL statement with user input may allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

	Factors: resultant to special character mismanagement, MAID, or blacklist/whitelist problems. Can be primary to authentication errors.
References	M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.
Node Relationships	ChildOf - Injection (74) ChildOf - Weaknesses in OWASP Top Ten (629) ChildOf - Weaknesses Examined by SAMATE (630) ChildOf - Weaknesses Used by NVD (635) ParentOf - SQL Injection: Hibernate (564)
Source Taxonomies	PLOVER - SQL injection 7 Pernicious Kingdoms - SQL Injection CLASP - SQL injection
Applicable Platforms	All

Path 1:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
```

```
...
143. ResultSet RS = Stmt.executeQuery("SELECT * FROM phpbb_users");
...
150. String MemberID = RS.getString("user_id");
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 2:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
```

```
...
150. String MemberID = RS.getString("user_id");
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

```
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 3:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
```

```
...
173. String MemberPostCount = RS.getString("user_posts");
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
```

```
...
203. String MemberSignature = Utils.wrapit(RS.getString("user_sig"));
```

▼

```
135. public static String wrapit(String text) //utils.java
```

```
...
144. strbuf.append(text.charAt(i));
...
148. return strbuf.toString();
```

▼

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
```

```
...
203. String MemberSignature = Utils.wrapit(RS.getString("user_sig"));
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
      228. String MemberHomepage = Utils.wrapit(RS.getString("user_website"));
```

```
135. public static String wrapit(String text) //utils.java
      ...
      144. strbuf.append(text.charAt(i));
      ...
      148. return strbuf.toString();
```

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
      228. String MemberHomepage = Utils.wrapit(RS.getString("user_website"));
      ...
      260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
      MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
      MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
      MemberCoolLink1 + "," + MemberCoolLink2 + ")";
      261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
      + st2;
      262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      ...
      343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 6:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
      229. String MemberYahoo = Utils.wrapit(RS.getString("user_yim"));
```

```
135. public static String wrapit(String text) //utils.java
      ...
      144. strbuf.append(text.charAt(i));
      ...
      148. return strbuf.toString();
```

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
      229. String MemberYahoo = Utils.wrapit(RS.getString("user_yim"));
      ...
```

```
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 7:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
230. String MemberAol = Utils.wrapit(RS.getString("user_aim"));
```

▼

```
135. public static String wrapit(String text) //utils.java
...
144. strbuf.append(text.charAt(i));
...
148. return strbuf.toString();
```

▼

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
230. String MemberAol = Utils.wrapit(RS.getString("user_aim"));
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 8:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
232. String MemberMsn = Utils.wrapit(RS.getString("user_msnm"));
```

▼

```
135. public static String wrapit(String text) //utils.java
```

```
...
144. strbuf.append(text.charAt(i));
...
148. return strbuf.toString();
```



```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
232. String MemberMsn = Utils.wrapit(RS.getString("user_msnm"));
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 9:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
236. String MemberCoolLink1 = Utils.wrapit(RS.getString("user_interests"));
```



```
135. public static String wrapit(String text) //utils.java
...
144. strbuf.append(text.charAt(i));
...
148. return strbuf.toString();
```



```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
...
236. String MemberCoolLink1 = Utils.wrapit(RS.getString("user_interests"));
...
260. String st2 = "(" + MemberID + "," + MemberViewCount + "," +
MemberPostCount + "," + MemberSignature + "," + MemberHomepage + "," +
MemberYahoo + "," + MemberAol + "," + MemberIcq + "," + MemberMsn + "," +
MemberCoolLink1 + "," + MemberCoolLink2 + ")";
261. String insertStatement = "insert into FORUM_PROFILE " + st1 + " values "
+ st2;
262. PreparedStatement ps = mvn.prepareStatement(insertStatement);
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
```

[Table of Contents] [Description]

Path 10:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      ...
      296. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_categories");
      ...
      302. String CategoryID = RS.getString("cat_id");
      ...
      330. CategoryID + "," +
      ...
      343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      344. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 11:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      ...
      302. String CategoryID = RS.getString("cat_id");
      ...
      330. CategoryID + "," +
      ...
      343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      344. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 12:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      ...
      304. String CategoryName = Utils.wrapit(RS.getString("cat_title"));
```

```
135. public static String wrapit(String text) //utils.java
      ...
      144. strbuf.append(text.charAt(i));
      ...
      148. return strbuf.toString();
```

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
      ...
      304. String CategoryName = Utils.wrapit(RS.getString("cat_title"));
      ...
      332. CategoryName + "," +
      ...
```

```
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
344. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 13:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
293. public static void migratecategories(Connection php, Connection mvn,
PrintStream ostream) throws SQLException //migrator.java
...
314. String CategoryOrder = RS.getString("cat_order");
...
336. CategoryOrder + "," +
...
343. PreparedStatement ps = mvn.prepareStatement(insertStatement);
344. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 14:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
360. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_forums");
...
371. String last_post_id = RS.getString("forum_last_post_id");
...
373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
phpbb_posts where post_id=" + last_post_id);
```

[Table of Contents] [Description]

Path 15:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
368. String ForumID = RS.getString("forum_id");
...
416. ForumID + ", " +
...
437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 16:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    369. String CategoryID = RS.getString("cat_id");
    ...
    417. CategoryID + ", " +
    ...
    437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
    438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 17:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    371. String last_post_id = RS.getString("forum_last_post_id");
    ...
    373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
    phpbb_posts where post_id=" + last_post_id);
```

[Table of Contents] [Description]

Path 18:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
    phpbb_posts where post_id=" + last_post_id);
    ...
    376. String poster_id = lastpostrs.getString("poster_id");
    ...
    382. ResultSet userstmtrs = userstmt.executeQuery("SELECT username from
    phpbb_users where user_id=" +poster_id);
```

[Table of Contents] [Description]

Path 19:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    376. String poster_id = lastpostrs.getString("poster_id");
    ...
    382. ResultSet userstmtrs = userstmt.executeQuery("SELECT username from
    phpbb_users where user_id=" +poster_id);
```

[Table of Contents] [Description]

Path 20:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
400. String ForumOrder = RS.getString("forum_order");
...
424. ForumOrder + ", " +
...
437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 21:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
404. String ForumStatus = RS.getString("forum_status");
...
428. ForumStatus + ", " +
...
437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 22:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
407. String ForumThreadCount = RS.getString("forum_topics");
...
431. ForumThreadCount + ", " +
...
437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 23:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        408. String ForumPostCount = RS.getString("forum_posts");
        ...
        432. ForumPostCount + ");";
        ...
        437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
        438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 24:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        483. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_posts");
        ...
        491. String ThreadID = RS.getString("topic_id");
        ...
        495. ResultSet ppidrs = ppidstmt.executeQuery("SELECT post_id from phpbb_posts
        where topic_id=" + ThreadID + " ORDER by post_id");
```

[Table of Contents] [Description]

Path 25:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        489. String PostID = RS.getString("post_id");
        ...
        525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
        phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 26:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        490. String ForumID = RS.getString("forum_id");
        ...
        580. ForumID + "," +
        ...
        602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
        603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 27:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        491. String ThreadID = RS.getString("topic_id");
        ...
        495. ResultSet ppidrs = ppidstmt.executeQuery("SELECT post_id from phpbb_posts
        where topic_id=" + ThreadID + " ORDER by post_id");
```

[Table of Contents] [Description]

Path 28:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        492. String MemberID = RS.getString("poster_id");
        ...
        513. ResultSet userstmtrs = userstmt.executeQuery("SELECT
        username,user_timezone from phpbb_users where user_id=" + MemberID);
```

[Table of Contents] [Description]

Path 29:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        495. ResultSet ppidrs = ppidstmt.executeQuery("SELECT post_id from phpbb_posts
        where topic_id=" + ThreadID + " ORDER by post_id");
        ...
        500. String ppidtmp = ppidrs.getString("post_id");
        ...
        504. ppid = ppidtmp;
        ...
        510. String ParentPostID = ppid;
        ...
        579. ParentPostID + "," +
        ...
        602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
        603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 30:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
500. String ppidtmp = ppidrs.getString("post_id");
...
504. ppid = ppidtmp;
...
510. String ParentPostID = ppid;
...
579. ParentPostID + "," +
...
602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 31:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
563. String PostEditCount = RS.getString("post_edit_count");
...
591. PostEditCount + "," +
...
602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 32:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
...
632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
where post_id = " + RS.getString("topic_last_post_id"));
...
635. LastPostMemberName = usernamefromid(php, lrs.getString("poster_id"));
```

▼

```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
    ...
107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
where user_id =" + id);
```

[Table of Contents] [Description]

Path 33:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
625. String ThreadID = RS.getString("topic_id");
...
725. ThreadID + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 34:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
627. String ForumID = RS.getString("forum_id");
...
726. ForumID + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 35:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
628. String MemberName = Utils.wrapit(usernamefromid(php, RS.getString
("topic_poster")));
```

```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
    ...
107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
where user_id =" + id);
```

[Table of Contents] [Description]

Path 36:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
        where post_id = " + RS.getString("topic_last_post_id"));
        ...
        635. LastPostMemberName = usernamefromid(php, lrs.getString("poster_id"));
```



```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
        ...
        107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
        where user_id =" + id);
```

[Table of Contents] [Description]

Path 37:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
        where post_id = " + RS.getString("topic_last_post_id"));
        ...
        635. LastPostMemberName = usernamefromid(php, lrs.getString("poster_id"));
```



```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
        ...
        107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
        where user_id =" + id);
```

[Table of Contents] [Description]

Path 38:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        635. LastPostMemberName = usernamefromid(php, lrs.getString("poster_id"));
```



```
103. public static String usernamefromid(Connection php, String id) throws
SQLException //migrator.java
        ...
        107. ResultSet mrs = mstmt.executeQuery("SELECT username from phpbb_users
        where user_id =" + id);
```

[Table of Contents] [Description]

Path 39:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
644. String first_post_id = RS.getString("topic_first_post_id");
...
646. lrs = lstmt.executeQuery("select post_text from phpbb_posts_text where
post_id = " + first_post_id);
```

[Table of Contents] [Description]

Path 40:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
665. String userID = RS.getString("topic_poster");
666. int timeZone = timeZoneFromId(php, userID);
```

▼

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 41:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
685. String last_post_id = RS.getString("topic_last_post_id");
...
687. lrs = lstmt.executeQuery("select post_time,poster_id from phpbb_posts
where post_id = " + last_post_id);
...
691. String userId = lrs.getString("poster_id");
692. int zone = timeZoneFromId(php, userId);
```

▼

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
```

```
    phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 42:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    687. lrs = lstmt.executeQuery("select post_time,poster_id from phpbb_posts
    where post_id = " + last_post_id);
    ...
    691. String userId = lrs.getString("poster_id");
    692. int zone = timeZoneFromId(php, userId);
```

▼

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
    525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
    phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 43:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    691. String userId = lrs.getString("poster_id");
    692. int zone = timeZoneFromId(php, userId);
```

▼

```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
    ...
    525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
    phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 44:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    706. String ThreadStatus = RS.getString("topic_status"); // TODO: Maybe remove
    status preservation
```

```
...
737. ThreadStatus + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 45:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
707. String ThreadHasPoll = RS.getString("topic_vote");
...
738. ThreadHasPoll + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 46:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
708. String ThreadViewCount = RS.getString("topic_views");
...
739. ThreadViewCount + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 47:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
...
709. String ThreadReplyCount = RS.getString("topic_replies");
...
740. ThreadReplyCount + "," +
...
748. PreparedStatement ps = mvn.prepareStatement(insertStatement);
749. ps.executeUpdate();
```

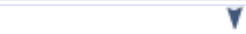
[Table of Contents] [Description]

Path 48:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
        ...
        665. String userID = RS.getString("topic_poster");
        666. int timeZone = timeZoneFromId(php, userID);
```



```
119. public static int timeZoneFromId(Connection php, String id) throws
SQLException //migrator.java
        ...
        525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
        phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 49:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        360. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_forums");
        ...
        371. String last_post_id = RS.getString("forum_last_post_id");
        ...
        373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
        phpbb_posts where post_id=" + last_post_id);
        ...
        376. String poster_id = lastpostrs.getString("poster_id");
        ...
        382. ResultSet userstmtrs = userstmt.executeQuery("SELECT username from
        phpbb_users where user_id=" +poster_id);
```

[Table of Contents] [Description]

Path 50:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        371. String last_post_id = RS.getString("forum_last_post_id");
        ...
        373. ResultSet lastpostrs = lastpoststmt.executeQuery("SELECT poster_id from
        phpbb_posts where post_id=" + last_post_id);
        ...
        376. String poster_id = lastpostrs.getString("poster_id");
        ...
```

```
382. ResultSet userstmtrs = userstmt.executeQuery("SELECT username from
phpbb_users where user_id=" +poster_id);
```

[Table of Contents] [Description]

Path 51:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
357. public static void migrateforums(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
360. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_forums");
    ...
400. String ForumOrder = RS.getString("forum_order");
    ...
424. ForumOrder + ", " +
    ...
437. PreparedStatement ps = mvn.prepareStatement(insertStatement);
438. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 52:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
483. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_posts");
    ...
492. String MemberID = RS.getString("poster_id");
    ...
513. ResultSet userstmtrs = userstmt.executeQuery("SELECT
username,user_timezone from phpbb_users where user_id=" + MemberID);
```

[Table of Contents] [Description]

Path 53:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
483. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_posts");
    ...
489. String PostID = RS.getString("post_id");
    ...
525. ResultSet posttextrs = posttextstmt.executeQuery("SELECT * from
phpbb_posts_text where post_id=" + PostID);
```

[Table of Contents] [Description]

Path 54:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      483. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_posts");
      ...
      492. String MemberID = RS.getString("poster_id");
      ...
      582. MemberID + "," +
      ...
      602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 55:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      489. String PostID = RS.getString("post_id");
      ...
      578. PostID + "," +
      ...
      602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 56:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
      ...
      491. String ThreadID = RS.getString("topic_id");
      ...
      581. ThreadID + "," +
      ...
      602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
      603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 57:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
480. public static void migrateposts(Connection php, Connection mvn, PrintStream
```

```
ostream) throws SQLException //migrator.java
    ...
    492. String MemberID = RS.getString("poster_id");
    ...
    582. MemberID + "," +
    ...
    602. PreparedStatement ps = mvn.prepareStatement(insertStatement);
    603. ps.executeUpdate();
```

[Table of Contents] [Description]

Path 58:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
    ...
    632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
    where post_id = " + RS.getString("topic_last_post_id"));
```

[Table of Contents] [Description]

Path 59:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    632. ResultSet lrs = lstmt.executeQuery("select poster_id from phpbb_posts
    where post_id = " + RS.getString("topic_last_post_id"));
```

[Table of Contents] [Description]

Path 60:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
    ...
    619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
    ...
    644. String first_post_id = RS.getString("topic_first_post_id");
    ...
    646. lrs = lstmt.executeQuery("select post_text from phpbb_posts_text where
    post_id = " + first_post_id);
```

[Table of Contents] [Description]

Path 61:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        619. ResultSet RS = Stmt.executeQuery("SELECT * from phpbb_topics");
        ...
        685. String last_post_id = RS.getString("topic_last_post_id");
        ...
        687. lrs = lstmt.executeQuery("select post_time,poster_id from phpbb_posts
        where post_id = " + last_post_id);
```

[Table of Contents] [Description]

Path 62:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        685. String last_post_id = RS.getString("topic_last_post_id");
        ...
        687. lrs = lstmt.executeQuery("select post_time,poster_id from phpbb_posts
        where post_id = " + last_post_id);
```

[Table of Contents] [Description]

Path 63:

Query Name - **Second_Order_SQL_Injection**

Severity - **Critical**

```
616. public static void migratethreads(Connection php, Connection mvn, PrintStream
ostream) throws SQLException //migrator.java
        ...
        749. ps.executeUpdate();
```

[Table of Contents] [Description]

Cross-site Scripting (XSS)

CWE ID	79
Description	A cross-site scripting weakness occurs when dynamically generated web pages display unvalidated, unfiltered, and unencoded user input allowing an attacker to embed malicious scripts into the generated page. This can be leveraged to execute scripting code as if it came from the site's server on the computer of anyone who uses the site.
Alternate Terms	"CSS" was once used as the acronym for this problem, but this can cause confusion with the "Cascading Style Sheets," so this acronym has declined significantly. Its use is discouraged by CWE.

Likelihood of Exploit	High to Very High
Weakness Ordinality	Resultant (Weakness is typically related to the presence of some other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Common Consequences	Confidentiality: The most common attack performed with cross-site scripting involves the disclosure of information stored in user cookies. Access control: In some circumstances it may be possible to run arbitrary code on a victim's computer when cross-site scripting is combined with other flaws. If successful, cross-site scripting vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems for a variety of nefarious purposes.
Potential Mitigations	Carefully check each input parameter against a rigorous positive specification (white list) defining the specific characters and format allowed. All input should be sanitized, not just parameters that the user is supposed to specify, but all data in the request, including hidden fields, cookies, headers, the URL itself, and so forth. A common mistake that leads to continuing XSS vulnerabilities is to validate only fields that are expected to be redisplayed by the site. We often encounter data from the request that is reflected by the application server or the application that the development team did not anticipate. Also, a field that is not currently reflected may be used by a future developer. Therefore, validating ALL parts of the HTTP request is recommended. This involves "HTML Entity Encoding" all non-alphanumeric characters from data that was received from the user and is now being written to the request. With Struts, you should write all data from form beans with the bean's filter attribute set to true. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as Internet Explorer), this attribute prevents the user's session cookie from being accessed by client-side scripts, including scripts inserted due to a XSS attack.
Demonstrative Examples	The following JSP code segment reads an employee ID, eid, from an HTTP request and displays it to the user. **JSP Example:** <% String eid = request.getParameter("eid"); %> ...

Employee ID: <%= eid %>

The following ASP.NET code segment reads an employee ID number from an HTTP request and displays it to the user.

ASP.NET Example:

```
protected System.Web.UI.WebControls.TextBox Login;
protected System.Web.UI.WebControls.Label EmployeeID;
  ...
EmployeeID.Text = Login.Text;
```

The code in this example operates correctly if eid contains only standard alphanumeric text. If eid has a value that includes meta-characters or source code, then the code will be executed by the web browser as it displays the HTTP response. Initially this might not appear to be much of a vulnerability. After all, why would someone enter a URL that causes malicious code to run on their own computer? The real danger is that an attacker will create the malicious URL, then use e-mail or social engineering tricks to lure victims into visiting a link to the URL. When victims click the link, they unwittingly reflect the malicious content through the vulnerable web application back to their own computers. This mechanism of exploiting vulnerable web applications is known as Reflected XSS.

The following JSP code segment queries a database for an employee with a given ID and prints the corresponding employee's name.

JSP Example:

```
<%...
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from emp
where id="+eid);
if (rs != null) {
rs.next();
String name = rs.getString("name");
%>
```

Employee Name: <%= name %>

The following ASP.NET code segment queries a database

for an employee with a given employee ID and prints the name corresponding with the ID.

ASP.NET Example:

```
protected System.Web.UI.WebControls.Label
EmployeeName;
  ...
string query = "select * from emp where id=" + eid;
sda = new SqlDataAdapter(query, conn);
sda.Fill(dt);
string name = dt.Rows[0]["Name"];
  ...
EmployeeName.Text = name;
```

This code functions correctly when the values of name are well-behaved, but it does nothing to prevent exploits if they are not. Again, this code can appear less dangerous because the value of name is read from a database, whose contents are apparently managed by the application. However, if the value of name originates from user-supplied data, then the database can be a conduit for malicious content. Without proper input validation on all data stored in the database, an attacker can execute malicious commands in the user's web browser. This type of exploit, known as Stored XSS, is particularly insidious because the indirection caused by the data store makes it more difficult to identify the threat and increases the possibility that the attack will affect multiple users. XSS got its start in this form with web sites that offered a "guestbook" to visitors. Attackers would include JavaScript in their guestbook entries, and all subsequent visitors to the guestbook page would execute the malicious code. As the examples demonstrate, XSS vulnerabilities are caused by code that includes unvalidated data in an HTTP response. There are three vectors by which an XSS attack can reach a victim: * As in the previous example, data is read directly from the HTTP request and reflected back in the HTTP response. Reflected XSS exploits occur when an attacker causes a user to supply dangerous content to a vulnerable web application, which is then reflected back to the user and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to victims. URLs constructed in this manner constitute the core of many phishing schemes, whereby an attacker convinces victims to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the user, the content is executed and proceeds to transfer private information, such as cookies that may include session information, from

the user's machine to the attacker or perform other nefarious activities. * As in this example, the application stores dangerous data in a database or other trusted data store. The dangerous data is subsequently read back into the application and included in dynamic content. Stored XSS exploits occur when an attacker injects dangerous content into a data store that is later read and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. * A source outside the application stores dangerous data in a database or other data store, and the dangerous data is subsequently read back into the application as trusted data and included in dynamic content.

Context Notes

Cross-site scripting (XSS) vulnerabilities occur when an attacker uses a web application to send malicious code, generally JavaScript, to a different end user. When a web application uses input from a user in the output it generates without filtering it, an attacker can insert an attack in that input and the web application sends the attack to other users. The end user trusts the web application, and the attacks exploit that trust to do things that would not normally be allowed. Attackers frequently use a variety of methods to encode the malicious portion of the tag, such as using Unicode, so the request looks less suspicious to the user. XSS attacks can generally be categorized into two categories: stored and reflected. Stored attacks are those where the injected code is permanently stored on the target servers in a database, message forum, visitor log, and so forth. Reflected attacks are those where the injected code takes another route to the victim, such as in an email message, or on some other server. When a user is tricked into clicking a link or submitting a form, the injected code travels to the vulnerable web server, which reflects the attack back to the user's browser. The browser then executes the code because it came from a 'trusted' server. For a reflected XSS attack to work, the victim must submit the attack to the server. This is still a very dangerous attack given the number of possible ways to trick a victim into submitting such a malicious request, including clicking a link on a malicious Web site, in an email, or in an inner-office posting. XSS flaws are very likely in web applications, as they require a great deal of developer discipline to avoid them in most applications. It is relatively easy for an attacker to find XSS vulnerabilities. Some of these vulnerabilities can be found using scanners, and some exist in older web application servers. The consequence of an XSS attack is the same regardless of whether it is

stored or reflected. The difference is in how the payload arrives at the server. XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. The most severe XSS attacks involve disclosure of the user's session cookie, which allows an attacker to hijack the user's session and take over their account. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirecting the user to some other page or site, and modifying presentation of content. Cross-site scripting (XSS) vulnerabilities occur when: 1. Data enters a Web application through an untrusted source, most frequently a web request. 2. The data is included in dynamic content that is sent to a web user without being validated for malicious code. The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data like cookies or other session information to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

Cross-site scripting attacks can occur wherever an untrusted user has the ability to publish content to a trusted web site. Typically, a malicious user will craft a client-side script, which -- when parsed by a web browser -- performs some activity (such as sending all site cookies to a given E-mail address). If the input is unchecked, this script will be loaded and run by each user visiting the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. There are several other possible attacks, such as running "Active X" controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy; cookie theft is however by far the most common. All of these attacks are easily prevented by ensuring that no script tags -- or for good measure, HTML tags at all -- are allowed in data to be posted publicly.

Cross-site scripting attacks may occur anywhere that possibly malicious users are allowed to post unregulated material to a trusted web site for the consumption of other valid users. The most common example can be found in bulletin-board web sites which provide web based mailing list-style functionality.

When used in conjunction with Cross-Site Request Forgery (CSRF), XSS worms can cause significant damage in web sites with large user populations.

References

Jeremiah Grossman, Robert "RSnake" Hansen, Petko "pdp" D. Petkov, Anton Rager and Seth Fogie. "XSS Attacks". Syngress. 2007.

M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.

Node Relationships

ChildOf - Injection (74)
ResultsIn - Mobile Code: Invoking Untrusted Mobile Code (494)
Peer - Cross-Site Request Forgery (CSRF) (352)
ChildOf - Weaknesses in OWASP Top Ten (629)
ChildOf - Weaknesses Used by NVD (635)
ParentOf - Basic XSS (80)
ParentOf - XSS in Error Pages (81)
ParentOf - Script in IMG Tags (82)
ParentOf - XSS using Script in Attributes (83)
ParentOf - XSS using Script Via Encoded URI Schemes (84)
ParentOf - Doubled Character XSS Manipulations (85)
ParentOf - Invalid Characters in Identifiers (86)
ParentOf - Alternate XSS Syntax (87)

Source Taxonomies

PLOVER - Cross-site scripting (XSS)

7 Pernicious Kingdoms - Cross-site Scripting

CLASP - Cross-site scripting

Applicable Platforms

All

Path 1:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
482. public static void saveVNTyperMode(GenericRequest request, GenericResponse
response) //myutil.java
      ...
485. saveVNTyperMode(request.getServletRequest(), response.getServletResponse
());
```

```
489. public static void saveVNTyperMode(HttpServletRequest request,
HttpServletResponse response) //myutil.java
      ...
491. String vnTyperMode = ParamUtil.getParameter(request, "vnselector");
```

```
100. public static String getParameter(HttpServletRequest request, String
param) //paramutil.java
      ...
102. String ret = request.getParameter(param);
...
104. return ret.trim();
```

```
489. public static void saveVNTyperMode(HttpServletRequest request,
HttpServletResponse response) //myutil.java
      ...
491. String vnTyperMode = ParamUtil.getParameter(request, "vnselector");
...
495. Cookie typerModeCookie = new Cookie(MVNForumConstant.VN_TYPER_MODE,
```

```
vnTyperMode);
496. typerModeCookie.setPath("/");
497. response.addCookie(typerModeCookie);
```

[Table of Contents] [Description]

Path 2:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

▼

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
134. String ret = getParameter(request, param, checkEmpty);
...
141. return ret;
```

▼

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
...
757. String fileName = ParamUtil.getParameterSafe(request, "filename",
true);// must check empty here
...
780. response.setHeader("Location", fileName);
```

[Table of Contents] [Description]

Path 3:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

▼

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
134. String ret = getParameter(request, param, checkEmpty);
```

```
...
141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
...
757. String fileName = ParamUtil.getParameterSafe(request, "filename",
true);// must check empty here
...
780. response.setHeader("Location", fileName);
781. response.setHeader("Content-Disposition", "attachment; filename=" +
fileName);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
134. String ret = getParameter(request, param, checkEmpty);
...
141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
...
757. String fileName = ParamUtil.getParameterSafe(request, "filename",
true);// must check empty here
...
780. response.setHeader("Location", fileName);
781. response.setHeader("Content-Disposition", "attachment; filename=" +
fileName);
...
784. response.setContentLength(length);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
```

```
boolean checkEmpty) //paramutil.java
    ...
    114. String ret = request.getParameter(param);
    ...
    116. ret = ret.trim();
    ...
    122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    134. String ret = getParameter(request, param, checkEmpty);
    ...
    141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
    ...
    757. String fileName = ParamUtil.getParameterSafe(request, "filename",
    true);// must check empty here
    ...
    780. response.setHeader("Location", fileName);
    781. response.setHeader("Content-Disposition", "attachment; filename=" +
    fileName);
    ...
    784. response.setContentLength(length);
    ...
    787. output = new BufferedOutputStream(response.getOutputStream(), 1024 /*
    buffer size */);
```

[Table of Contents] [Description]

Path 6:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    114. String ret = request.getParameter(param);
    ...
    116. ret = ret.trim();
    ...
    122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    134. String ret = getParameter(request, param, checkEmpty);
    ...
    141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
```

```
...
757. String fileName = ParamUtil.getParameterSafe(request, "filename",
true);// must check empty here
...
780. response.setHeader("Location", fileName);
781. response.setHeader("Content-Disposition", "attachment; filename=" +
fileName);
...
784. response.setContentLength(length);
...
788. response.setBufferSize(1024);
```

[Table of Contents] [Description]

Path 7:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
532. Cookie[] cookies = request.getCookies();
...
535. Cookie cookie = cookies[i];
...
540. encodedPassword = cookie.getValue();
...
548. user = login(request, null, memberName, encodedPassword, true);
```

▼

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
210. encodedPassword = memberPassword;
...
237. manageAutoLogin(memberName, encodedPassword, request, response);
```

▼

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
...
316. Cookie passwordCookie = new Cookie(MVNFORUM_COOKIE_PASSWORD,
encodedPassword);
317. passwordCookie.setMaxAge(autoLoginExpire);
318. passwordCookie.setPath(MVNFORUM_COOKIE_PATH);
...
321. response.addCookie(passwordCookie);
```

[Table of Contents] [Description]

Path 8:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
1. <%-- //listpendinguserssuccess.jsp
...
79. <td><b>&raquo; </b><a class="command" href="<%=urlResolver.encodeURL
```

```
(request, response, "listpendingusers?offset=" + request.getParameter
("Offset"))%>"><fmt:message
key="mvnforum.admin.success.return_to_listpending_user"/></a> (<fmt:message
key="mvnforum.common.success.automatic"/>)</td>
```

[Table of Contents] [Description]

Path 9:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    540. encodedPassword = cookie.getValue();
    ...
    548. user = login(request, null, memberName, encodedPassword, true);
```

▼

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    210. encodedPassword = memberPassword;
    ...
    237. manageAutoLogin(memberName, encodedPassword, request, response);
```

▼

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
    ...
    316. Cookie passwordCookie = new Cookie(MVNFORUM_COOKIE_PASSWORD,
    encodedPassword);
    317. passwordCookie.setMaxAge(autoLoginExpire);
    318. passwordCookie.setPath(MVNFORUM_COOKIE_PATH);
    ...
    321. response.addCookie(passwordCookie);
```

[Table of Contents] [Description]

Path 10:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
100. public static String getParameter(HttpServletRequest request, String
param) //paramutil.java
    ...
    102. String ret = request.getParameter(param);
    ...
    104. return ret.trim();
```

▼

```
489. public static void saveVNTyperMode(HttpServletRequest request,
HttpServletResponse response) //myutil.java
    ...
    491. String vnTyperMode = ParamUtil.getParameter(request, "vnselector");
    ...
```

```
495. Cookie typerModeCookie = new Cookie(MVNForumConstant.VN_TYPER_MODE,
vnTyperMode);
496. typerModeCookie.setPath("/");
497. response.addCookie(typerModeCookie);
```

[Table of Contents] [Description]

Path 11:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

▼

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
...
133. String memberPasswordMD5 = ParamUtil.getParameter(request, "md5pw",
false);
...
140. processLogin(request, response, memberName, memberPassword,
memberPasswordMD5);
```

▼

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
147. String memberName, String memberPassword, String memberPasswordMD5)
...
169. user = login(request, response, memberName, memberPasswordMD5, true);
```

▼

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
210. encodedPassword = memberPassword;
...
237. manageAutoLogin(memberName, encodedPassword, request, response);
```

▼

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
...
316. Cookie passwordCookie = new Cookie(MVNFORUM_COOKIE_PASSWORD,
encodedPassword);
317. passwordCookie.setMaxAge(autoLoginExpire);
318. passwordCookie.setPath(MVNFORUM_COOKIE_PATH);
...
321. response.addCookie(passwordCookie);
```

[Table of Contents] [Description]

Path 12:

Query Name - **Reflected_XSS_All_Clients**

Severity - **Critical**

```
292. public static String getParameterPassword(HttpServletRequest request, String
param, int minLength, int option) //paramutil.java
        ...
        296. String ret = request.getParameter(param);
        ...
        298. ret = ret.trim();
        ...
        312. return ret;
```

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
        ...
        136. memberPassword = ParamUtil.getParameterPassword(request, "MemberMatkhau",
        3, 0);
        ...
        140. processLogin(request, response, memberName, memberPassword,
        memberPasswordMD5);
```

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
        147. String memberName, String memberPassword, String memberPasswordMD5)
        ...
        166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
        194. String memberName, String memberPassword, boolean passwordEncoded)
        ...
        210. encodedPassword = memberPassword;
        ...
        237. manageAutoLogin(memberName, encodedPassword, request, response);
```

```
297. protected void manageAutoLogin(String memberName, String
encodedPassword, //onlineusermanager.java
        ...
        316. Cookie passwordCookie = new Cookie(MVNFORUM_COOKIE_PASSWORD,
        encodedPassword);
        317. passwordCookie.setMaxAge(autoLoginExpire);
        318. passwordCookie.setPath(MVNFORUM_COOKIE_PATH);
        ...
        321. response.addCookie(passwordCookie);
```

[Table of Contents] [Description]

Cross-site Scripting (XSS)

CWE ID	79
Description	A cross-site scripting weakness occurs when dynamically generated web pages display unvalidated, unfiltered, and unencoded user input allowing an attacker to embed

	malicious scripts into the generated page. This can be leveraged to execute scripting code as if it came from the site's server on the computer of anyone who uses the site.
Alternate Terms	"CSS" was once used as the acronym for this problem, but this can cause confusion with the "Cascading Style Sheets," so this acronym has declined significantly. Its use is discouraged by CWE.
Likelihood of Exploit	High to Very High
Weakness Ordinality	Resultant (Weakness is typically related to the presence of some other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Common Consequences	Confidentiality: The most common attack performed with cross-site scripting involves the disclosure of information stored in user cookies. Access control: In some circumstances it may be possible to run arbitrary code on a victim's computer when cross-site scripting is combined with other flaws. If successful, cross-site scripting vulnerabilities can be exploited to manipulate or steal cookies, create requests that can be mistaken for those of a valid user, compromise confidential information, or execute malicious code on the end user systems for a variety of nefarious purposes.
Potential Mitigations	Carefully check each input parameter against a rigorous positive specification (white list) defining the specific characters and format allowed. All input should be sanitized, not just parameters that the user is supposed to specify, but all data in the request, including hidden fields, cookies, headers, the URL itself, and so forth. A common mistake that leads to continuing XSS vulnerabilities is to validate only fields that are expected to be redisplayed by the site. We often encounter data from the request that is reflected by the application server or the application that the development team did not anticipate. Also, a field that is not currently reflected may be used by a future developer. Therefore, validating ALL parts of the HTTP request is recommended. This involves "HTML Entity Encoding" all non-alphanumeric characters from data that was received from the user and is now being written to the request. With Struts, you should write all data from form beans with the bean's filter attribute set to true. To help mitigate XSS attacks against the user's session cookie, set the session cookie to be HttpOnly. In browsers that support the HttpOnly feature (such as Internet Explorer), this attribute prevents the user's session cookie from being accessed by client-side scripts, including scripts inserted due to a XSS attack.
Demonstrative Examples	The following JSP code segment reads an employee ID, eid, from an HTTP request and displays it to the user.

JSP Example:

```
<% String eid = request.getParameter("eid"); %>
...
Employee ID: <%= eid %>
```

The following ASP.NET code segment reads an employee ID number from an HTTP request and displays it to the user.

ASP.NET Example:

```
protected System.Web.UI.WebControls.TextBox Login;
protected System.Web.UI.WebControls.Label EmployeeID;
  ...
EmployeeID.Text = Login.Text;
```

The code in this example operates correctly if eid contains only standard alphanumeric text. If eid has a value that includes meta-characters or source code, then the code will be executed by the web browser as it displays the HTTP response. Initially this might not appear to be much of a vulnerability. After all, why would someone enter a URL that causes malicious code to run on their own computer? The real danger is that an attacker will create the malicious URL, then use e-mail or social engineering tricks to lure victims into visiting a link to the URL. When victims click the link, they unwittingly reflect the malicious content through the vulnerable web application back to their own computers. This mechanism of exploiting vulnerable web applications is known as Reflected XSS.

The following JSP code segment queries a database for an employee with a given ID and prints the corresponding employee's name.

JSP Example:

```
<%...
Statement stmt = conn.createStatement();
ResultSet rs = stmt.executeQuery("select * from emp
where id="+eid);
if (rs != null) {
rs.next();
```

```
String name = rs.getString("name");
%>

Employee Name: <%= name %>
```

The following ASP.NET code segment queries a database for an employee with a given employee ID and prints the name corresponding with the ID.

ASP.NET Example:

```
protected System.Web.UI.WebControls.Label
EmployeeName;
  ...
string query = "select * from emp where id=" + eid;
sda = new SqlDataAdapter(query, conn);
sda.Fill(dt);
string name = dt.Rows[0]["Name"];
  ...
EmployeeName.Text = name;
```

This code functions correctly when the values of name are well-behaved, but it does nothing to prevent exploits if they are not. Again, this code can appear less dangerous because the value of name is read from a database, whose contents are apparently managed by the application. However, if the value of name originates from user-supplied data, then the database can be a conduit for malicious content. Without proper input validation on all data stored in the database, an attacker can execute malicious commands in the user's web browser. This type of exploit, known as Stored XSS, is particularly insidious because the indirection caused by the data store makes it more difficult to identify the threat and increases the possibility that the attack will affect multiple users. XSS got its start in this form with web sites that offered a "guestbook" to visitors. Attackers would include JavaScript in their guestbook entries, and all subsequent visitors to the guestbook page would execute the malicious code. As the examples demonstrate, XSS vulnerabilities are caused by code that includes unvalidated data in an HTTP response. There are three vectors by which an XSS attack can reach a victim: * As in the previous example, data is read directly from the HTTP request and reflected back in the HTTP response. Reflected XSS exploits occur when an attacker causes a user to supply dangerous content to a vulnerable web application, which is then reflected back to the user and executed by the web browser. The most common mechanism for delivering malicious content is to include it as a parameter in a URL that is posted publicly or e-mailed directly to victims. URLs constructed in this

	manner constitute the core of many phishing schemes, whereby an attacker convinces victims to visit a URL that refers to a vulnerable site. After the site reflects the attacker's content back to the user, the content is executed and proceeds to transfer private information, such as cookies that may include session information, from the user's machine to the attacker or perform other nefarious activities. * As in this example, the application stores dangerous data in a database or other trusted data store. The dangerous data is subsequently read back into the application and included in dynamic content. Stored XSS exploits occur when an attacker injects dangerous content into a data store that is later read and included in dynamic content. From an attacker's perspective, the optimal place to inject malicious content is in an area that is displayed to either many users or particularly interesting users. Interesting users typically have elevated privileges in the application or interact with sensitive data that is valuable to the attacker. If one of these users executes malicious content, the attacker may be able to perform privileged operations on behalf of the user or gain access to sensitive data belonging to the user. * A source outside the application stores dangerous data in a database or other data store, and the dangerous data is subsequently read back into the application as trusted data and included in dynamic content.
Context Notes	Cross-site scripting (XSS) vulnerabilities occur when an attacker uses a web application to send malicious code, generally JavaScript, to a different end user. When a web application uses input from a user in the output it generates without filtering it, an attacker can insert an attack in that input and the web application sends the attack to other users. The end user trusts the web application, and the attacks exploit that trust to do things that would not normally be allowed. Attackers frequently use a variety of methods to encode the malicious portion of the tag, such as using Unicode, so the request looks less suspicious to the user. XSS attacks can generally be categorized into two categories: stored and reflected. Stored attacks are those where the injected code is permanently stored on the target servers in a database, message forum, visitor log, and so forth. Reflected attacks are those where the injected code takes another route to the victim, such as in an email message, or on some other server. When a user is tricked into clicking a link or submitting a form, the injected code travels to the vulnerable web server, which reflects the attack back to the user's browser. The browser then executes the code because it came from a 'trusted' server. For a reflected XSS attack to work, the victim must submit the attack to the server. This is still a very dangerous attack given the number of possible ways to trick a victim into submitting such a malicious request, including clicking a link on a malicious Web site, in an email, or in an inner-office

posting. XSS flaws are very likely in web applications, as they require a great deal of developer discipline to avoid them in most applications. It is relatively easy for an attacker to find XSS vulnerabilities. Some of these vulnerabilities can be found using scanners, and some exist in older web application servers. The consequence of an XSS attack is the same regardless of whether it is stored or reflected. The difference is in how the payload arrives at the server. XSS can cause a variety of problems for the end user that range in severity from an annoyance to complete account compromise. The most severe XSS attacks involve disclosure of the user's session cookie, which allows an attacker to hijack the user's session and take over their account. Other damaging attacks include the disclosure of end user files, installation of Trojan horse programs, redirecting the user to some other page or site, and modifying presentation of content. Cross-site scripting (XSS) vulnerabilities occur when: 1. Data enters a Web application through an untrusted source, most frequently a web request. 2. The data is included in dynamic content that is sent to a web user without being validated for malicious code. The malicious content sent to the web browser often takes the form of a segment of JavaScript, but may also include HTML, Flash or any other type of code that the browser may execute. The variety of attacks based on XSS is almost limitless, but they commonly include transmitting private data like cookies or other session information to the attacker, redirecting the victim to web content controlled by the attacker, or performing other malicious operations on the user's machine under the guise of the vulnerable site.

Cross-site scripting attacks can occur wherever an untrusted user has the ability to publish content to a trusted web site. Typically, a malicious user will craft a client-side script, which -- when parsed by a web browser -- performs some activity (such as sending all site cookies to a given E-mail address). If the input is unchecked, this script will be loaded and run by each user visiting the web site. Since the site requesting to run the script has access to the cookies in question, the malicious script does also. There are several other possible attacks, such as running "Active X" controls (under Microsoft Internet Explorer) from sites that a user perceives as trustworthy; cookie theft is however by far the most common. All of these attacks are easily prevented by ensuring that no script tags -- or for good measure, HTML tags at all -- are allowed in data to be posted publicly.

Cross-site scripting attacks may occur anywhere that possibly malicious users are allowed to post unregulated material to a trusted web site for the consumption of other valid users. The most common example can be found in bulletin-board web sites which provide web based mailing list-style functionality.

	When used in conjunction with Cross-Site Request Forgery (CSRF), XSS worms can cause significant damage in web sites with large user populations.
References	Jeremiah Grossman, Robert "RSnake" Hansen, Petko "pdp" D. Petkov, Anton Rager and Seth Fogie. "XSS Attacks". Syngress. 2007. M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.
Node Relationships	ChildOf - Injection (74) ResultsIn - Mobile Code: Invoking Untrusted Mobile Code (494) Peer - Cross-Site Request Forgery (CSRF) (352) ChildOf - Weaknesses in OWASP Top Ten (629) ChildOf - Weaknesses Used by NVD (635) ParentOf - Basic XSS (80) ParentOf - XSS in Error Pages (81) ParentOf - Script in IMG Tags (82) ParentOf - XSS using Script in Attributes (83) ParentOf - XSS using Script Via Encoded URI Schemes (84) ParentOf - Doubled Character XSS Manipulations (85) ParentOf - Invalid Characters in Identifiers (86) ParentOf - Alternate XSS Syntax (87)
Source Taxonomies	PLOVER - Cross-site scripting (XSS) 7 Pernicious Kingdoms - Cross-site Scripting CLASP - Cross-site scripting
Applicable Platforms	All

Path 1:

Query Name - **Stored_XSS**

Severity - **Critical**

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
      151. String MemberName = RS.getString("username");
      ...
      161. MemberName = Utils.wrapit(MemberName);
```

```
135. public static String wrapit(String text) //utils.java
      ...
      144. strbuf.append(text.charAt(i));
      ...
      146. strbuf.append('\'');
      ...
      148. return strbuf.toString();
```

```
137. public static void migrateusers(Connection php, Connection mvn, PrintStream
ostream) throws SQLException, FileNotFoundException //migrator.java
      ...
```

```
161. MemberName = Utils.wrapit(MemberName);
...
283. System.out.println(userCount + ":" + MemberName + " updated!");
```

[Table of Contents] [Description]

Path 2:

Query Name - **Stored_XSS**

Severity - **Critical**

```
196. public AttachmentBean getAttachment(int attachID) //attachmentdaoimpljdbc.java
...
222. bean.setAttachMimeType(resultSet.getString("AttachMimeType"));
```

▼

```
106. public void setAttachMimeType(String attachMimeType) //attachmentbean.java
107. this.attachMimeType = StringUtil.getEmptyStringIfNull(attachMimeType);
```

▼

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

▼

```
106. public void setAttachMimeType(String attachMimeType) //attachmentbean.java
107. this.attachMimeType = StringUtil.getEmptyStringIfNull(attachMimeType);
```

▼

```
196. public AttachmentBean getAttachment(int attachID) //attachmentdaoimpljdbc.java
...
222. bean.setAttachMimeType(resultSet.getString("AttachMimeType"));
...
231. return bean;
```

▼

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
...
608. attachBean = DAOFactory.getAttachmentDAO().getAttachment(attachID);
...
650. response.setContentType(attachBean.getAttachMimeType());
```

▼

```
103. public String getAttachMimeType() //attachmentbean.java
104. return attachMimeType;
```

▼

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
...
650. response.setContentType(attachBean.getAttachMimeType());
```

[Table of Contents] [Description]

Path 3:

Query Name - **Stored_XSS**

Severity - **Critical**

```
196. public AttachmentBean getAttachment(int attachID) //attachmentdaoimpljdbc.java
     ...
     222. bean.setAttachMimeType(resultSet.getString("AttachMimeType"));
```

```
106. public void setAttachMimeType(String attachMimeType) //attachmentbean.java
     107. this.attachMimeType = StringUtil.getEmptyStringIfNull(attachMimeType);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
     123. return str;
```

```
106. public void setAttachMimeType(String attachMimeType) //attachmentbean.java
     107. this.attachMimeType = StringUtil.getEmptyStringIfNull(attachMimeType);
```

```
196. public AttachmentBean getAttachment(int attachID) //attachmentdaoimpljdbc.java
     ...
     222. bean.setAttachMimeType(resultSet.getString("AttachMimeType"));
     ...
     231. return bean;
```

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
     ...
     608. attachBean = DAOFactory.getAttachmentDAO().getAttachment(attachID);
     ...
     651. response.setHeader("Location", attachBean.getAttachFilename());
```

```
89. public String getAttachFilename() //attachmentbean.java
     90. return attachFilename;
```

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
     ...
     651. response.setHeader("Location", attachBean.getAttachFilename());
```

[Table of Contents] [Description]

Path 4:

Query Name - **Stored_XSS**

Severity - **Critical**

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
     ...
     261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
```

```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
      99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
      ...
      123. return str;
```



```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
      99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```



```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
      ...
      261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
      ...
      269. return bean;
```



```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
      ...
      418. pmAttachBean = DAOFactory.getPmAttachmentDAO().getPmAttachment(attachID);
      ...
      465. response.setContentType(pmAttachBean.getPmAttachMimeType());
      466. response.setHeader("Location", pmAttachBean.getPmAttachFilename());
```



```
81. public String getPmAttachFilename() //pmattachmentbean.java
      82. return pmAttachFilename;
```



```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
      ...
      466. response.setHeader("Location", pmAttachBean.getPmAttachFilename());
      ...
      471. response.setHeader("Cache-Control", "max-age=" + cacheTime);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Stored_XSS**

Severity - **Critical**

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
      ...
      261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
```



```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
      99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
     123. return str;
```

```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
     99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
     ...
     261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
     ...
     269. return bean;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
     418. pmAttachBean = DAOFactory.getPmAttachmentDAO().getPmAttachment(attachID);
     ...
     465. response.setContentType(pmAttachBean.getPmAttachMimeType());
     ...
     474. response.setHeader("Content-Disposition", "attachment; filename=" +
     pmAttachBean.getPmAttachFilename());
```

```
81. public String getPmAttachFilename() //pmattachmentbean.java
     82. return pmAttachFilename;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
     ...
     474. response.setHeader("Content-Disposition", "attachment; filename=" +
     pmAttachBean.getPmAttachFilename());
```

[Table of Contents] [Description]

Path 6:

Query Name - **Stored_XSS**

Severity - **Critical**

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
     ...
     261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
```

```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
     99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
```

```
123. return str;
```

```
98. public void setPmAttachMimeType(String pmAttachMimeType) //pmattachmentbean.java
99. this.pmAttachMimeType = StringUtil.getEmptyStringIfNull(pmAttachMimeType);
```

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java
...
261. bean.setPmAttachMimeType(resultSet.getString("PmAttachMimeType"));
...
269. return bean;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
...
418. pmAttachBean = DAOFactory.getPmAttachmentDAO().getPmAttachment(attachID);
...
465. response.setContentType(pmAttachBean.getPmAttachMimeType());
...
474. response.setHeader("Content-Disposition", "attachment; filename=" +
pmAttachBean.getPmAttachFilename());
```

```
81. public String getPmAttachFilename() //pmattachmentbean.java
82. return pmAttachFilename;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
...
474. response.setHeader("Content-Disposition", "attachment; filename=" +
pmAttachBean.getPmAttachFilename());
...
480. outputStream = response.getOutputStream();
```

[Table of Contents] [Description]

Log Forging

CWE ID	117
Description	Writing unvalidated user input into log files can allow an attacker to forge log entries or inject malicious content into logs.
Likelihood of Exploit	Medium
Weakness Ordinality	Primary (Weakness exists independent of other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Potential Mitigations	Assume all input is malicious. Use an appropriate combination of black lists and white lists to ensure only valid, expected and appropriate input is processed by the system.
Demonstrative Examples	The following web application code attempts to read an integer value from a request object. If the value fails to parse as an integer, then the input is logged with an error message

indicating what happened.

```
...
String val = request.getParameter("val");
 try {
  int value = Integer.parseInt(val);
 }
 catch (NumberFormatException) {
  log.info("Failed to parse val = " + val);
 }
...
```

If a user submits the string "twenty-one" for val, the following entry is logged: INFO: Failed to parse val=twenty-one However, if an attacker submits the string "twenty-one%0a%0aINFO:+User+logged+out%3dbadguy", the following entry is logged: INFO: Failed to parse val=twenty-one INFO: User logged out=badguy Clearly, attackers can use this same mechanism to insert arbitrary log entries.

Context Notes Log forging vulnerabilities occur when: 1. Data enters an application from an untrusted source. 2. The data is written to an application or system log file. Applications typically use log files to store a history of events or transactions for later review, statistics gathering, or debugging. Depending on the nature of the application, the task of reviewing log files may be performed manually on an as-needed basis or automated with a tool that automatically culls logs for important events or trending information. Interpretation of the log files may be hindered or misdirected if an attacker can supply data to the application that is subsequently logged verbatim. In the most benign case, an attacker may be able to insert false entries into the log file by providing the application with input that includes appropriate characters. If the log file is processed automatically, the attacker can render the file unusable by corrupting the format of the file or injecting unexpected characters. A more subtle attack might involve skewing the log file statistics. Forged or otherwise, corrupted log files can be used to cover an attacker's tracks or even to implicate another party in the commission of a malicious act [22]. In the worst case, an attacker may inject code or other commands into the log file and take advantage of a vulnerability in the log processing utility [17].

References [17] G. Hoglund and G. McGraw. "Exploiting Software: How to Break Code". Addison-Wesley. February 2004.

[22] A. Muffet. "The night the log was forged". <http://doc.novsu.ac.ru/oreilly/tcpip/puis/ch10_05.htm>.

Node Relationships ChildOf - Output Validation (116)

Source Taxonomies 7 Pernicious Kingdoms - Log Forging

Applicable Platforms All

Path 1:

Query Name - **Log_Forgery**

Severity - **Warning**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      114. String ret = request.getParameter(param);
      ...
      116. ret = ret.trim();
      ...
      122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      134. String ret = getParameter(request, param, checkEmpty);
      ...
      141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
      ...
      757. String fileName = ParamUtil.getParameterSafe(request, "filename",
      true);// must check empty here
      ...
      769. File file = new File(MVNForumConfig.getLogDir() + File.separatorChar +
      fileName);
      ...
      771. log.error("Can't find a file " + file + " to be downloaded (or maybe it's
      directory).");
```

[Table of Contents] [Description]

Path 2:

Query Name - **Log_Forgery**

Severity - **Warning**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      114. String ret = request.getParameter(param);
      ...
      116. ret = ret.trim();
      ...
      122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      134. String ret = getParameter(request, param, checkEmpty);
      ...
      141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
```

```
response) //generaladmintaskswebhandler.java
    ...
    757. String fileName = ParamUtil.getParameterSafe(request, "filename",
    true);// must check empty here
    ...
    794. log.error("Can't find the such log file on server " + fileName);
```

[Table of Contents] [Description]

Path 3:

Query Name - **Log_Forgery**

Severity - **Warning**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    114. String ret = request.getParameter(param);
    ...
    116. ret = ret.trim();
    ...
    122. return ret;
```

▼

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    134. String ret = getParameter(request, param, checkEmpty);
    ...
    141. return ret;
```

▼

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
    ...
    757. String fileName = ParamUtil.getParameterSafe(request, "filename",
    true);// must check empty here
    ...
    796. log.error("Error while trying to send backup file from server (" +
    fileName + ").", e);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Log_Forgery**

Severity - **Warning**

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
    ...
    149. String[] removeList = request.getParameterValues("remove");
    ...
    152. log.debug("perm = " + removeList[i]);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Log_Forgery**

Severity - **Warning**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      128. String[] removeList = request.getParameterValues("remove");
      ...
      131. log.debug("perm = " + removeList[i]);
```

[Table of Contents] [Description]

Path 6:

Query Name - **Log_Forgery**

Severity - **Warning**

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
      ...
      258. String[] removeList = request.getParameterValues("remove");
      ...
      261. log.debug("perm = " + removeList[i]);
```

[Table of Contents] [Description]

Path 7:

Query Name - **Log_Forgery**

Severity - **Warning**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
      ...
      131. String[] removeList = request.getParameterValues("remove");
      ...
      134. log.debug("perm = " + removeList[i]);
```

[Table of Contents] [Description]

Path 8:

Query Name - **Log_Forgery**

Severity - **Warning**

```
1214. public static void deleteMember(MemberBean memberBean, boolean
deleteSentMessages, GenericRequest request, Locale locale) //memberwebhandler.java
      ...
      1246. bufferPicFile.append(request.getRealPath
      (MVNForumGlobal.UPLOADED_AVATAR_DIR));
      1247. bufferPicFile.append(File.separatorChar).append(memberName).append
      (".jpg");
      1248. String picFile = bufferPicFile.toString();
      ...
      1250. log.debug("Delete avatar = " + picFile);
```

[Table of Contents] [Description]

Path 9:

Query Name - **Log_Forgery**

Severity - **Warning**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
      ...
      95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      150. onlineUser.reloadPermission();
```

```
179. public void reloadPermission() //onlineuserimpl.java
      ...
      189. log.error("Error when reload permission in OnlineUserImpl for memberID =
" + memberID, ex);
```

[Table of Contents] [Description]

Path 10:

Query Name - **Log_Forgery**

Severity - **Warning**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

▼

```
631. public void prepareView_forCurrentMember(GenericRequest
request) //memberwebhandler.java
      544. String memberEmail = GenericParamUtil.getParameterEmail(request,
      "MemberEmail");
      ...
      634. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      639. onlineUser.updateNewMessageCount(true);
```

▼

```
215. public boolean updateNewMessageCount(boolean forceUpdate) //onlineuserimpl.java
      ...
      235. log.error("Error when udpate new message count in OnlineUserImpl for
      memberID = " + memberID, ex);
```

[Table of Contents] [Description]

Path 11:

Query Name - **Log_Forgery**

Severity - **Warning**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      893. String memberEmail = GenericParamUtil.getParameter(request,
      "MemberEmail");
      ...
      988. mailMessageStruct.setTo(memberEmail);
```

```
62. public void setTo(String to) //mailmessagestruct.java
      63. this.to = to;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      988. mailMessageStruct.setTo(memberEmail);
      ...
      992. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      193. mailList.add(mailItem);
      ...
      195. sendMail(mailList);
      ...
      199. log.debug("from = " + mailItem.getFrom());
```

```
52. public String getFrom() //mailmessagestruct.java
      53. return from;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      199. log.debug("from = " + mailItem.getFrom());
```

[Table of Contents] [Description]

Path 12:

Query Name - **Log_Forgery**

Severity - **Warning**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
```

```
79. return ret;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
     ...
     118. String subject = GenericParamUtil.getParameter(request, "Subject", true);
     ...
     139. mailMessageStruct.setSubject(subject);
```

```
83. public void setSubject(String subject) //mailmessagestruct.java
     84. this.subject = subject;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
     ...
     139. mailMessageStruct.setSubject(subject);
     ...
     143. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
     ...
     193. mailList.add(mailItem);
     ...
     195. sendMail(mailList);
     ...
     200. log.debug("to = " + mailItem.getTo());
```

```
59. public String getTo() //mailmessagestruct.java
     60. return to;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
     ...
     200. log.debug("to = " + mailItem.getTo());
```

[Table of Contents] [Description]

Path 13:

Query Name - **Log_Forgery**

Severity - **Warning**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
     ...
     71. String ret = request.getParameter(param);
     ...
     73. ret = ret.trim();
     ...
     79. return ret;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
     ...
     118. String subject = GenericParamUtil.getParameter(request, "Subject", true);
```

```
...
139. mailMessageStruct.setSubject(subject);
```

```
83. public void setSubject(String subject) //mailmessagestruct.java
    84. this.subject = subject;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
    ...
    139. mailMessageStruct.setSubject(subject);
    ...
    143. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
    ...
    193. mailList.add(mailItem);
    ...
    195. sendMail(mailList);
    ...
    201. log.debug("cc = " + mailItem.getCc());
```

```
66. public String getCc() //mailmessagestruct.java
    67. return cc;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
    ...
    201. log.debug("cc = " + mailItem.getCc());
```

[Table of Contents] [Description]

Path 14:

Query Name - **Log_Forgery**

Severity - **Warning**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
    ...
    118. String subject = GenericParamUtil.getParameter(request, "Subject", true);
    ...
    139. mailMessageStruct.setSubject(subject);
```

```
83. public void setSubject(String subject) //mailmessagestruct.java
    84. this.subject = subject;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
      ...
      139. mailMessageStruct.setSubject(subject);
      ...
      143. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      193. mailList.add(mailItem);
      ...
      195. sendMail(mailList);
      ...
      202. log.debug("bcc = " + mailItem.getBcc());
```

```
73. public String getBcc() //mailmessagestruct.java
      74. return bcc;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      202. log.debug("bcc = " + mailItem.getBcc());
```

[Table of Contents] [Description]

Path 15:

Query Name - **Log_Forgery**

Severity - **Warning**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
      ...
      118. String subject = GenericParamUtil.getParameter(request, "Subject", true);
      ...
      139. mailMessageStruct.setSubject(subject);
```

```
83. public void setSubject(String subject) //mailmessagestruct.java
      84. this.subject = subject;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
      ...
      139. mailMessageStruct.setSubject(subject);
      ...
      143. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      193. mailList.add(mailItem);
      ...
      195. sendMail(mailList);
      ...
      203. log.debug("subject = " + mailItem.getSubject());
```

```
80. public String getSubject() //mailmessagestruct.java
      81. return subject;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
      ...
      203. log.debug("subject = " + mailItem.getSubject());
```

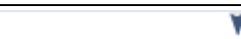
[Table of Contents] [Description]

Path 16:

Query Name - **Log_Forgery**

Severity - **Warning**

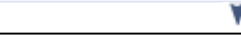
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
      ...
      95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      150. onlineUser.reloadPermission();
```



```
179. public void reloadPermission() //onlineuserimpl.java
      ...
      189. log.error("Error when reload permission in OnlineUserImpl for memberID =
```

```
" + memberID, ex);
```

[Table of Contents] [Description]

Path 17:

Query Name - **Log_Forgery**

Severity - **Warning**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

▼

```
631. public void prepareView_forCurrentMember(GenericRequest
request) //memberwebhandler.java
    544. String memberEmail = GenericParamUtil.getParameterEmail(request,
    "MemberEmail");
    ...
    634. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    639. onlineUser.updateNewMessageCount(true);
```

▼

```
215. public boolean updateNewMessageCount(boolean forceUpdate) //onlineuserimpl.java
    ...
    235. log.error("Error when udpate new message count in OnlineUserImpl for
    memberID = " + memberID, ex);
```

[Table of Contents] [Description]

Path 18:

Query Name - **Log_Forgery**

Severity - **Warning**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
```

```
...
73. ret = ret.trim();
...
79. return ret;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
...
118. String subject = GenericParamUtil.getParameter(request, "Subject", true);
...
139. mailMessageStruct.setSubject(subject);
```

```
83. public void setSubject(String subject) //mailmessagestruct.java
84. this.subject = subject;
```

```
106. public void sendEmailProcess(GenericRequest request) //mailwebhandler.java
...
139. mailMessageStruct.setSubject(subject);
...
143. MailUtil.sendMail(mailMessageStruct);
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
...
193. mailList.add(mailItem);
...
195. sendMail(mailList);
...
199. log.debug("from = " + mailItem.getFrom());
```

```
52. public String getFrom() //mailmessagestruct.java
53. return from;
```

```
189. public static void sendMail(MailMessageStruct mailItem) //mailutil.java
...
199. log.debug("from = " + mailItem.getFrom());
```

[Table of Contents] [Description]

Password Misuse

CWE ID	10011
Description	Passwords should not be retrieved from the database.
Likelihood of Exploit	Low
Common Consequences	The password may be leaked to an unauthorized party.
Potential Mitigations	Do not retrieve passwords from the database.
Applicable Platforms	All

Path 1:

Query Name - **Password_Misuse**

Severity - **Warning**

```
550. public MemberBean getMemberFromMemberName(String memberName) throws
ObjectNotFoundException, DatabaseException //memberdaoimpljdbc.java
        ...
563.    resultSet = statement.executeQuery();
        ...
571.    bean.setMemberPassword(resultSet.getString("MemberPassword"));
```

```
530. public void setMemberPassword(String memberPassword) //memberbean.java
```

[Table of Contents] [Description]

Path 2:

Query Name - **Password_Misuse**

Severity - **Warning**

```
106. public Collection getBeans() throws DatabaseException //phpbbuserdaoimpl.java
        ...
120.    resultSet = statement.executeQuery();
        ...
126.    bean.setUserPassword(resultSet.getString("user_password"));
```

```
391. public void setUserPassword(String userPassword) //phpbbusers.java
```

[Table of Contents] [Description]

Path 3:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
        ...
905.    resultSet = statement.executeQuery();
        ...
910.    String retValue = resultSet.getString("MemberPassword");
        ...
912.    return retValue;
```

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
        ...
571.    String encryptedPassword = memberDAO.getPassword(memberID);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
```

```
...
953. resultSet = statement.executeQuery();
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

[Table of Contents] [Description]

Path 5:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1297. public int getNumberOfMembers() //memberdaoimpljdbc.java
...
1310. resultSet = statement.executeQuery();
...
1312. return resultSet.getInt(1);
```

```
247. public int getNumberOfMembers() throws
DatabaseException //memberdaoexternaluserdatabaseabstract.java
248. return memberDAO.getNumberOfMembers();
```

```
74. public void prepareList(GenericRequest request, GenericResponse response, String
requestURI) //forumwebhandler.java
...
121. int numberOfMembers = DAOFactory.getMemberDAO().getNumberOfMembers();
122. request.setAttribute("NumberOfMembers", new Integer(numberOfMembers));
...
127. request.setAttribute("OnlineUserActions",
onlineUserManager.getOnlineUserActions(0 /*default*/, duplicateUsers)); // no
permission
...
135. CategoryTreeListener treelistener = new CategoryTreePath(request,
response, CategoryTreePath.FORUMID_IN_LIST_CATEGORY_CASE, categoryId, null);
```

```
100. public CategoryTreePath(GenericRequest request, GenericResponse response, int
forumID, int categoryID, String display) //categorytreepath.java
```

```
...
108. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
532. Cookie[] cookies = request.getCookies();
...
535. Cookie cookie = cookies[i];
...
540. encodedPassword = cookie.getValue();
```

[Table of Contents] [Description]

Path 6:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java
...
109. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
256. return memberDAO.getPassword(memberID);
```

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
...
904. statement.setInt(1, memberID);
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
256. return memberDAO.getPassword(memberID);
```

```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java
...
```

```
110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
```

[Table of Contents] [Description]

Path 7:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
      ...
      1447. resultSet = statement.executeQuery();
      ...
      1451. return resultSet.getInt(1);
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
      (memberEmail);
      ...
      936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
```

▼

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
```

▼

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
      ...
      123. return str;
```

▼

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      958. return StringUtil.getEmptyStringIfNull(resultSet.getString
      ("MemberTempPassword"));
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
```

[Table of Contents] [Description]

Path 8:

Query Name - **Password_Misuse**

Severity - **Warning**

```
978. public int getNumberOfPosts_inThread(int threadID, boolean
enable) //postdaoimpljdbc.java
      ...
```

```
999. resultSet = statement.executeQuery();
...
1001. return resultSet.getInt(1);
```

```
966. public int getNumberOfEnablePosts_inThread(int threadID) //postdaoimpljdbc.java
...
969. return getNumberOfPosts_inThread(threadID, true);
```

```
278. public void prepareDelete(GenericRequest request, GenericResponse
response) //threadwebhandler.java
...
296. int numberOfPosts = DAOFactory.getPostDAO
().getNumberOfEnablePosts_inThread(threadID);
...
348. request.setAttribute("NumberOfPosts", new Integer(numberOfPosts));
349. request.setAttribute("NumberOfPendingPosts", new Integer
(numberOfPendingPosts));
...
357. CategoryTreeListener treelistener = new CategoryTreePath(request,
response, forumID, null, null, stb.toString());
```

```
117. public CategoryTreePath(GenericRequest request, GenericResponse response, int
forumID, String mode, String action, String display) //categorytreepath.java
...
126. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
532. Cookie[] cookies = request.getCookies();
...
535. Cookie cookie = cookies[i];
...
540. encodedPassword = cookie.getValue();
```

[Table of Contents] [Description]

Path 9:

Query Name - **Password_Misuse**

Severity - **Warning**

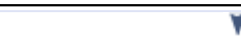
```
2178. public int getNumberOfEnableThreadsWithPendingPosts() //threaddaoimpljdbc.java
...
2191. resultSet = statement.executeQuery();
...
2193. return resultSet.getInt(1);
```

```
1150. public void prepareModerationControlPanel(GenericRequest request,
GenericResponse response) //threadwebhandler.java
...
1182. int threadsWithPendingPostsCount = DAOFactory.getThreadDAO
().getNumberOfEnableThreadsWithPendingPosts();
...
```

```
1188. request.setAttribute("ThreadsWithPendingPostsCount", new Integer
(threadsWithPendingPostsCount));
...
1192. CategoryTreeListener listener = categoryService.getManagementModcp
(request, response);
```

```
129. public CategoryTreeListener getManagementModcp(GenericRequest request,
GenericResponse response) //categoryserviceimpldefault.java
...
132. return new ModcpImpl(request, response);
```

```
94. public ModcpImpl(GenericRequest request, GenericResponse
response) //modcpimpl.java
...
99. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
532. Cookie[] cookies = request.getCookies();
...
535. Cookie cookie = cookies[i];
...
540. encodedPassword = cookie.getValue();
```

[Table of Contents] [Description]

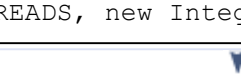
Path 10:

Query Name - **Password_Misuse**

Severity - **Warning**

```
2204. public int getNumberOfEnableThreadsWithPendingPosts_inForum(int
forumID) //threaddaoimpljdbc.java
...
2218. resultSet = statement.executeQuery();
...
2220. return resultSet.getInt(1);
```

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
...
878. threadsWithPendingPostsCount = DAOFactory.getThreadDAO
().getNumberOfEnableThreadsWithPendingPosts_inForum(forumID);
...
898. request.setAttribute("ThreadsWithPendingPostsCount", new Integer
(threadsWithPendingPostsCount));
...
904. request.setAttribute("Result", tree.build());
...
907. OnlineUserUtil.setRequestAttributeOfOnlineActions(request,
Action.PAGE_ID_LISTTHREADS, new Integer(forumID));
```



```
58. public static void setRequestAttributeOfOnlineActions(GenericRequest request,
```

```
int pageID, Object pageParam) //onlineuserutil.java
      ...
      64. onlineUserManager.getOnlineUser(request);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      532. Cookie[] cookies = request.getCookies();
      ...
      535. Cookie cookie = cookies[i];
      ...
      540. encodedPassword = cookie.getValue();
```

[Table of Contents] [Description]

Path 11:

Query Name - **Password_Misuse**

Severity - **Warning**

```
550. public MemberBean getMemberFromMemberName(String memberName) throws
ObjectNotFoundException, DatabaseException //memberdaoimpljdbc.java
      ...
      563. resultSet = statement.executeQuery();
      ...
      571. bean.setMemberPassword(resultSet.getString("MemberPassword"));
```



```
530. public void setMemberPassword(String memberPassword) //memberbean.java
      531. this.memberPassword = memberPassword;
```

[Table of Contents] [Description]

Path 12:

Query Name - **Password_Misuse**

Severity - **Warning**

```
550. public MemberBean getMemberFromMemberName(String memberName) throws
ObjectNotFoundException, DatabaseException //memberdaoimpljdbc.java
      ...
      563. resultSet = statement.executeQuery();
      ...
      571. bean.setMemberPassword(resultSet.getString("MemberPassword"));
```



```
530. public void setMemberPassword(String memberPassword) //memberbean.java
      531. this.memberPassword = memberPassword;
```

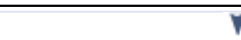
[Table of Contents] [Description]

Path 13:

Query Name - **Password_Misuse**

Severity - **Warning**

```
106. public Collection getBeans() throws DatabaseException //phpbbuserdaoimpl.java
     ...
     120. resultSet = statement.executeQuery();
     ...
     126. bean.setUserPassword(resultSet.getString("user_password"));
```



```
391. public void setUserPassword(String userPassword) //phpbbusers.java
     392. this.userPassword = userPassword;
```

[Table of Contents] [Description]

Path 14:

Query Name - **Password_Misuse**

Severity - **Warning**

```
106. public Collection getBeans() throws DatabaseException //phpbbuserdaoimpl.java
     ...
     120. resultSet = statement.executeQuery();
     ...
     126. bean.setUserPassword(resultSet.getString("user_password"));
```



```
391. public void setUserPassword(String userPassword) //phpbbusers.java
     392. this.userPassword = userPassword;
```

[Table of Contents] [Description]

Path 15:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
     905. resultSet = statement.executeQuery();
     ...
     910. String retValue = resultSet.getString("MemberPassword");
     ...
     912. return retValue;
```



```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
     ...
     571. String encryptedPassword = memberDAO.getPassword(memberID);
     572. if (encryptedPassword.length() > 16) {
```

[Table of Contents] [Description]

Path 16:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
```

```
...
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
...
571. String encryptedPassword = memberDAO.getPassword(memberID);
...
573. return encryptedPassword.substring(0, 16);
```

[Table of Contents] [Description]

Path 17:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
...
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
...
571. String encryptedPassword = memberDAO.getPassword(memberID);
...
575. if (encryptedPassword.length() < 16) {
```

[Table of Contents] [Description]

Path 18:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
...
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
...
571. String encryptedPassword = memberDAO.getPassword(memberID);
...
```

```
576. for (int i = encryptedPassword.length(); i < 16; i++) {
```

[Table of Contents] [Description]

Path 19:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

▼

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
     ...
571. String encryptedPassword = memberDAO.getPassword(memberID);
     ...
577. encryptedPassword = encryptedPassword + " ";
```

[Table of Contents] [Description]

Path 20:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

▼

```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java
     ...
571. String encryptedPassword = memberDAO.getPassword(memberID);
     ...
577. encryptedPassword = encryptedPassword + " ";
```

[Table of Contents] [Description]

Path 21:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
```

```
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```



```
569. public static String getConfirmedEmailKey (int memberID) throws
ObjectNotFoundException, DatabaseException //myutil.java

...
571. String encryptedPassword = memberDAO.getPassword(memberID);
...
577. encryptedPassword = encryptedPassword + " ";
...
580. return encryptedPassword;
```

[Table of Contents] [Description]

Path 22:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java

...
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```



```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java

...
110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
```

[Table of Contents] [Description]

Path 23:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java

...
905. resultSet = statement.executeQuery();
...
910. String retValue = resultSet.getString("MemberPassword");
...
912. return retValue;
```



```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java

...
110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
...
114. return password.equals(pass);
```

[Table of Contents] [Description]

Path 24:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     256. return memberDAO.getPassword(memberID);
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
     ...
613. String currentPassword = DAOFactory.getMemberDAO().getPassword(memberID);
```

[Table of Contents] [Description]

Path 25:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     256. return memberDAO.getPassword(memberID);
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
     ...
613. String currentPassword = DAOFactory.getMemberDAO().getPassword(memberID);
614. if (currentPassword.equals(memberPassword)) {
```

[Table of Contents] [Description]

Path 26:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     256. return memberDAO.getPassword(memberID);
```

```
1312. public void processUpdateCMSPassword(GenericRequest request, String
memberCurrentMakhau, String memberPassword1, String
memberPassword2) //memberwebhandler.java
     ...
1333. String currentPassword = DAOFactory.getMemberDAO().getPassword
(memberID);
```

[Table of Contents] [Description]

Path 27:

Query Name - **Password_Misuse**

Severity - **Warning**

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
     ...
905. resultSet = statement.executeQuery();
     ...
910. String retValue = resultSet.getString("MemberPassword");
     ...
912. return retValue;
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     256. return memberDAO.getPassword(memberID);
```

```
1312. public void processUpdateCMSPassword(GenericRequest request, String
memberCurrentMakhau, String memberPassword1, String
memberPassword2) //memberwebhandler.java
     ...
1333. String currentPassword = DAOFactory.getMemberDAO().getPassword
(memberID);
1334. if (currentPassword.equals(memberPassword)) {
```

[Table of Contents] [Description]

Path 28:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     953. resultSet = statement.executeQuery();
     ...
     958. return StringUtil.getEmptyStringIfNull(resultSet.getString
     ("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
     123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     958. return StringUtil.getEmptyStringIfNull(resultSet.getString
     ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
     936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
     (memberID);
     ...
     941. if (currentTempPassword.length() < 5) {
```

[Table of Contents] [Description]

Path 29:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     953. resultSet = statement.executeQuery();
     ...
     958. return StringUtil.getEmptyStringIfNull(resultSet.getString
     ("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
     123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     958. return StringUtil.getEmptyStringIfNull(resultSet.getString
     ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
     936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
     (memberID);
     ...
     948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
```

```
"UTF-8");
```

[Table of Contents] [Description]

Path 30:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
953. resultSet = statement.executeQuery();
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
"UTF-8");
```

[Table of Contents] [Description]

Path 31:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
953. resultSet = statement.executeQuery();
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
```

```
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
"UTF-8");
...
958. passwordResetUrl.append(urlEncodedTempPassword);
```

[Table of Contents] [Description]

Path 32:

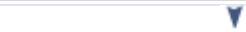
Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
953. resultSet = statement.executeQuery();
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```



```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
971. root.put("currentTempPassword", currentTempPassword);
```

[Table of Contents] [Description]

Path 33:

Query Name - **Password_Misuse**

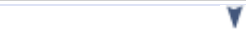
Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
953. resultSet = statement.executeQuery();
```

```
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

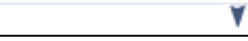
```
121. public static String getEmptyStringIfNull(String str) //stringutil.java

...
123. return str;
```



```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java

...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```



```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java

...
1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

[Table of Contents] [Description]

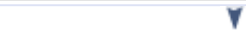
Path 34:

Query Name - **Password_Misuse**

Severity - **Warning**

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java

...
953. resultSet = statement.executeQuery();
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

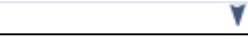


```
121. public static String getEmptyStringIfNull(String str) //stringutil.java

...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java

...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```



```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java

...
1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
1025. if (memberTempPassword.equals(currentTempPassword) == false) {
```

[Table of Contents] [Description]

Path 35:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1297. public int getNumberOfMembers() //memberdaoimpljdbc.java
      ...
      1310. resultSet = statement.executeQuery();
      ...
      1312. return resultSet.getInt(1);
```

```
247. public int getNumberOfMembers() throws
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      248. return memberDAO.getNumberOfMembers();
```

```
74. public void prepareList(GenericRequest request, GenericResponse response, String
requestURI) //forumwebhandler.java
      ...
      121. int numberOfMembers = DAOFactory.getMemberDAO().getNumberOfMembers();
      122. request.setAttribute("NumberOfMembers", new Integer(numberOfMembers));
      ...
      127. request.setAttribute("OnlineUserActions",
      onlineUserManager.getOnlineUserActions(0 /*default*/, duplicateUsers)); // no
      permission
      ...
      135. CategoryTreeListener treelistener = new CategoryTreePath(request,
      response, CategoryTreePath.FORUMID_IN_LIST_CATEGORY_CASE, categoryId, null);
```

```
100. public CategoryTreePath(GenericRequest request, GenericResponse response, int
forumID, int categoryID, String display) //categorytreepath.java
      ...
      108. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      532. Cookie[] cookies = request.getCookies();
      ...
      535. Cookie cookie = cookies[i];
      ...
      540. encodedPassword = cookie.getValue();
      ...
      546. (encodedPassword.equalsIgnoreCase(PASSWORD_OF_METHOD_CUSTOMIZATION) ==
      false) ) {
```

[Table of Contents] [Description]

Path 36:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      1417. resultSet = statement.executeQuery();
      ...
      1421. return resultSet.getInt(1);
```

```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java
    ...
    109. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (memberName);
    110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
```



```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    256. return memberDAO.getPassword(memberID);
```



```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
    ...
    904. statement.setInt(1, memberID);
    905. resultSet = statement.executeQuery();
    ...
    910. String retValue = resultSet.getString("MemberPassword");
    ...
    912. return retValue;
```



```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    256. return memberDAO.getPassword(memberID);
```



```
105. public boolean isCorrectCurrentPassword(String memberName, String password,
boolean encoded) //samplerealmauthenticator.java
    ...
    110. String pass = DAOFactory.getMemberDAO().getPassword(memberID);
    ...
    114. return password.equals(pass);
```

[Table of Contents] [Description]

Path 37:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    1417. resultSet = statement.executeQuery();
    ...
    1421. return resultSet.getInt(1);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
    141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (memberName);
    ...
    154. return memberID;
```



```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
```

```
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

[Table of Contents] [Description]

Path 38:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
...
154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
```

```
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
941. if (currentTempPassword.length() < 5) {
```

[Table of Contents] [Description]

Path 39:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
...
154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
```

```
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
"UTF-8");
```

[Table of Contents] [Description]

Path 40:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
...
154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
       123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
       ...
       958. return StringUtil.getEmptyStringIfNull(resultSet.getString
       ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
       ...
       936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
       (memberID);
       ...
       948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
       "UTF-8");
```

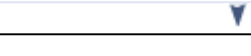
[Table of Contents] [Description]

Path 41:

Query Name - **Password_Misuse**

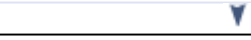
Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
       ...
       1417. resultSet = statement.executeQuery();
       ...
       1421. return resultSet.getInt(1);
```

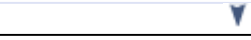


```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
       ...
       141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
       (memberName);
       ...
       154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
       ...
       901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
       (memberName);
       ...
       936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
       (memberID);
```



```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
       ...
       999. statement.setInt(1, memberID);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
```

```
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     958. return StringUtil.getEmptyStringIfNull(resultSet.getString
     ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
     936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
     (memberID);
     ...
     948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
     "UTF-8");
     ...
     958. passwordResetUrl.append(urlEncodedTempPassword);
```

[Table of Contents] [Description]

Path 42:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
     ...
     1417. resultSet = statement.executeQuery();
     ...
     1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
     ...
     141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
     (memberName);
     ...
     154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
     901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
     (memberName);
     ...
     936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
     (memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
     ...
     999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
     123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
        ...
        958. return StringUtil.getEmptyStringIfNull(resultSet.getString
        ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
        ...
        936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
        (memberID);
        ...
        971. root.put("currentTempPassword", currentTempPassword);
```

[Table of Contents] [Description]

Path 43:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
        ...
        1417. resultSet = statement.executeQuery();
        ...
        1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
        ...
        141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
        (memberName);
        ...
        154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
        ...
        1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
        (memberName);
        ...
        1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
        (memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
        ...
        999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
        ...
        123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
        ...
```

```
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
...
1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

[Table of Contents] [Description]

Path 44:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
...
154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
...
1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
```

```
...
1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
1025. if (memberTempPassword.equals(currentTempPassword) == false) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
...
1447. resultSet = statement.executeQuery();
...
1451. return resultSet.getInt(1);
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
(memberEmail);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

▼

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

▼

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

▼

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
941. if (currentTempPassword.length() < 5) {
```

[Table of Contents] [Description]

Path 46:

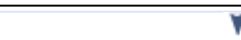
Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
      ...
      1447. resultSet = statement.executeQuery();
      ...
      1451. return resultSet.getInt(1);
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
      (memberEmail);
      ...
      936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
      ...
      123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      958. return StringUtil.getEmptyStringIfNull(resultSet.getString
      ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
      ...
      948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
      "UTF-8");
```

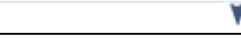
[Table of Contents] [Description]

Path 47:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
      ...
      1447. resultSet = statement.executeQuery();
      ...
      1451. return resultSet.getInt(1);
```



```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
```

```
910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
(memberEmail);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
958. return StringUtil.getEmptyStringIfNull(resultSet.getString
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
...
948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
"UTF-8");
```

[Table of Contents] [Description]

Path 48:

Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
...
1447. resultSet = statement.executeQuery();
...
1451. return resultSet.getInt(1);
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
(memberEmail);
...
936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
(memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
       123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
       ...
       958. return StringUtil.getEmptyStringIfNull(resultSet.getString
       ("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
       ...
       936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
       (memberID);
       ...
       948. String urlEncodedTempPassword = URLEncoder.encode(currentTempPassword,
       "UTF-8");
       ...
       958. passwordResetUrl.append(urlEncodedTempPassword);
```

[Table of Contents] [Description]

Path 49:

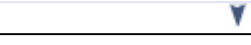
Query Name - **Password_Misuse**

Severity - **Warning**

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
       ...
       1447. resultSet = statement.executeQuery();
       ...
       1451. return resultSet.getInt(1);
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
       ...
       910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
       (memberEmail);
       ...
       936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
       (memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
       ...
       999. statement.setInt(1, memberID);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
       123. return str;
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
       ...
       958. return StringUtil.getEmptyStringIfNull(resultSet.getString
```

```
("MemberTempPassword"));
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
    ...
    936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
    (memberID);
    ...
    971. root.put("currentTempPassword", currentTempPassword);
```

[Table of Contents] [Description]

Path 50:

Query Name - **Password_Misuse**

Severity - **Warning**

```
978. public int getNumberOfPosts_inThread(int threadID, boolean
enable) //postdaoimpljdbc.java
    ...
    999. resultSet = statement.executeQuery();
    ...
    1001. return resultSet.getInt(1);
```

```
966. public int getNumberOfEnablePosts_inThread(int threadID) //postdaoimpljdbc.java
    ...
    969. return getNumberOfPosts_inThread(threadID, true);
```

```
278. public void prepareDelete(GenericRequest request, GenericResponse
response) //threadwebhandler.java
    ...
    296. int numberOfPosts = DAOFactory.getPostDAO
    ().getNumberOfEnablePosts_inThread(threadID);
    ...
    348. request.setAttribute("NumberOfPosts", new Integer(numberOfPosts));
    349. request.setAttribute("NumberOfPendingPosts", new Integer
    (numberOfPendingPosts));
    ...
    357. CategoryTreeListener treelistener = new CategoryTreePath(request,
    response, forumID, null, null, stb.toString());
```

```
117. public CategoryTreePath(GenericRequest request, GenericResponse response, int
forumID, String mode, String action, String display) //categorytreepath.java
    ...
    126. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    532. Cookie[] cookies = request.getCookies();
    ...
    535. Cookie cookie = cookies[i];
    ...
    540. encodedPassword = cookie.getValue();
    ...
```

```
546. (encodedPassword.equalsIgnoreCase(PASSWORD_OF_METHOD_CUSTOMIZATION) ==
false) ) {
```

[Table of Contents] [Description]

Path 51:

Query Name - **Password_Misuse**

Severity - **Warning**

```
2178. public int getNumberOfEnableThreadsWithPendingPosts() //threaddaoimpljdbc.java
      ...
      2191. resultSet = statement.executeQuery();
      ...
      2193. return resultSet.getInt(1);
```

```
1150. public void prepareModerationControlPanel(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
      1182. int threadsWithPendingPostsCount = DAOFactory.getThreadDAO
      ().getNumberOfEnableThreadsWithPendingPosts();
      ...
      1188. request.setAttribute("ThreadsWithPendingPostsCount", new Integer
      (threadsWithPendingPostsCount));
      ...
      1192. CategoryTreeListener listener = categoryService.getManagementModcp
      (request, response);
```

```
129. public CategoryTreeListener getManagementModcp(GenericRequest request,
GenericResponse response) //categoryserviceimpldefault.java
      ...
      132. return new ModcpImpl(request, response);
```

```
94. public ModcpImpl(GenericRequest request, GenericResponse
response) //modcpimpl.java
      ...
      99. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      532. Cookie[] cookies = request.getCookies();
      ...
      535. Cookie cookie = cookies[i];
      ...
      540. encodedPassword = cookie.getValue();
      ...
      545. (encodedPassword.equalsIgnoreCase(PASSWORD_OF_METHOD_REALM) == false) &&
```

[Table of Contents] [Description]

Path 52:

Query Name - **Password_Misuse**

Severity

- **Warning**

```
2178. public int getNumberOfEnableThreadsWithPendingPosts() //threaddaoimpljdbc.java
      ...
      2191. resultSet = statement.executeQuery();
      ...
      2193. return resultSet.getInt(1);
```

```
1150. public void prepareModerationControlPanel(GenericRequest request,
GenericResponse response) //threadwebhandler.java
      ...
      1182. int threadsWithPendingPostsCount = DAOFactory.getThreadDAO
      ().getNumberOfEnableThreadsWithPendingPosts();
      ...
      1188. request.setAttribute("ThreadsWithPendingPostsCount", new Integer
      (threadsWithPendingPostsCount));
      ...
      1192. CategoryTreeListener listener = categoryService.getManagementModcp
      (request, response);
```

```
129. public CategoryTreeListener getManagementModcp(GenericRequest request,
GenericResponse response) //categoryserviceimpldefault.java
      ...
      132. return new ModcpImpl(request, response);
```

```
94. public ModcpImpl(GenericRequest request, GenericResponse
response) //modcpimpl.java
      ...
      99. onlineUser = OnlineUserManager.getInstance().getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      532. Cookie[] cookies = request.getCookies();
      ...
      535. Cookie cookie = cookies[i];
      ...
      540. encodedPassword = cookie.getValue();
      ...
      546. (encodedPassword.equalsIgnoreCase(PASSWORD_OF_METHOD_CUSTOMIZATION) ==
      false) ) {
```

[Table of Contents] [Description]

Path 53:

Query Name - **Password_Misuse**

Severity - **Warning**

```
2204. public int getNumberOfEnableThreadsWithPendingPosts_inForum(int
forumID) //threaddaoimpljdbc.java
      ...
      2218. resultSet = statement.executeQuery();
      ...
      2220. return resultSet.getInt(1);
```

```
819. public void prepareList_limit(GenericRequest request, GenericResponse response,
String requestURI) //threadwebhandler.java
    ...
    878. threadsWithPendingPostsCount = DAOFactory.getThreadDAO
    ().getNumberOfEnableThreadsWithPendingPosts_inForum(forumID);
    ...
    898. request.setAttribute("ThreadsWithPendingPostsCount", new Integer
    (threadsWithPendingPostsCount));
    ...
    904. request.setAttribute("Result", tree.build());
    ...
    907. OnlineUserUtil.setRequestAttributeOfOnlineActions(request,
    Action.PAGE_ID_LISTTHREADS, new Integer(forumID));
```

```
58. public static void setRequestAttributeOfOnlineActions(GenericRequest request,
int pageID, Object pageParam) //onlineuserutil.java
    ...
    64. onlineUserManager.getOnlineUser(request);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    532. Cookie[] cookies = request.getCookies();
    ...
    535. Cookie cookie = cookies[i];
    ...
    540. encodedPassword = cookie.getValue();
    ...
    546. (encodedPassword.equalsIgnoreCase(PASSWORD_OF_METHOD_CUSTOMIZATION) ==
    false) ) {
```

[Table of Contents] [Description]

Improper Resource Shutdown or Release

CWE ID	404
Description	The program can potentially fail to release or incorrectly release a system resource. A resource is not properly cleared and made available for re-use.
Functional Area	Non-specific
Potential Mitigations	It is good practice to be responsible for freeing all resources you allocate. Memory should be allocated/freed using matching functions such as malloc/free, new/delete, and new[]/delete[].
Demonstrative Examples	The following method never closes the file handle it opens. The Finalize() method for StreamReader eventually calls Close (), but there is no guarantee as to how long it will take before the Finalize() method is invoked. In fact, there is no guarantee that Finalize() will ever be invoked. In a busy environment, this can result in the VM using up all of its available file handles.

```
private void processFile(string fName) {
  StreamWriter sw = new
  StreamWriter(fName);
  string line;
  while ((line = sr.ReadLine()) != null) processLine(line);
}
```

If an exception occurs after establishing the database connection and before the same connection closes, the pool of database connections may become exhausted. If the number of available connections is exceeded, other users cannot access this resource, effectively denying access to the application. Using the following database connection pattern will ensure that all opened connections are closed. The con.close() call should be the first executable statement in the finally block.

```
try {
  Connection con = DriverManager.getConnection
(some_connection_string)
}
catch ( Exception e ) {
  log( e )
}
finally {
  con.close()
}
```

Under normal conditions the following C# code executes a database query, processes the results returned by the database, and closes the allocated SqlConnection object. But if an exception occurs while executing the SQL or processing the results, the SqlConnection object is not closed. If this happens often enough, the database will run out of available cursors and not be able to execute any more SQL queries.

C# Example:

```
...
SqlConnection conn = new SqlConnection(connString);
SqlCommand cmd = new SqlCommand(queryString);
cmd.Connection = conn; conn.Open();
SqlDataReader rdr = cmd.ExecuteReader();
HarvestResults(rdr);
conn.Connection.Close();
...
```

The following C function does not close the file handle it opens if an error occurs. If the process is long-lived, the

process can run out of file handles.

C Example:

```
int decodeFile(char* fName) {
  char buf[BUF_SZ];
  FILE* f = fopen(fName, "r");
  if (!f) {
    printf("cannot open %s\n", fName);
    return DECODE_FAIL;
  }
  else {
    while (fgets(buf, BUF_SZ, f)) {
      if (!checkChecksum(buf)) {
        return DECODE_FAIL;
      }
      else {
        decodeBlock(buf);
      }
    }
  }
  fclose(f);
  return DECODE_SUCCESS;
}
```

In this example, the program fails to use matching functions such as malloc/free, new/delete, and new[]/delete[] to allocate/deallocate the resource.

C++ Example:

```
class A{
  void foo();
};
void A::foo(){
  int *ptr;
  ptr = (int*)malloc(sizeof(int));
  delete ptr;
}
```

In this example, the program calls the delete[] function on non-heap memory.

C++ Example:

```
class A{
  void foo(bool);
```

```
};
void A::foo(bool heap) {
  int localArray[2] = {11,22};
  int *p = localArray;
  if (heap){
    p = new int[2];
  }
  delete[] p;
}
```

Observed Examples

CVE-1999-1127 - Does not shut down named pipe connections if malformed data is sent.

CVE-2001-0830 - Sockets not properly closed when attacker repeatedly connects and disconnects from server.

CVE-2002-1372 - Return values of file/socket operations not checked, allowing resultant consumption of file descriptors.

Context Notes

It is important to be consistent with how and where you free memory in a function. If you allocate memory that you intend to free upon completion of the function, you must be sure to free the memory at all exit points for that function including error conditions.

Before freeing a pointer, the programmer should make sure that the pointer was previously allocated and that the memory belongs to the programmer. Freeing an unallocated pointer will cause undefined behavior in the program.

Can be resultant from improper error handling or insufficient resource tracking.

Overlaps memory leaks, asymmetric resource consumption, malformed input errors.

Most unreleased resource issues result in general software reliability problems, but if an attacker can intentionally trigger a resource leak, the attacker might be able to launch a denial of service attack by depleting the resource pool. Resource leaks have at least two common causes: - Error conditions and other exceptional circumstances. - Confusion over which part of the program is responsible for releasing the resource.

Node Relationships

ChildOf - Resource Management Errors (399)
Peer - Memory Leak (401)
Peer - Asymmetric Resource Consumption (Amplification) (405)

Source Taxonomies

PLOVER - Improper resource shutdown or release

7 Pernicious Kingdoms - Unreleased Resource

Applicable Platforms All

Path 1:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
186. public static FileInputStream getInput() //dbutils.java
        ...
188. FileInputStream file = new FileInputStream
(Phpbb2MvnforumConfig.EXPORT_XML);
```

[Table of Contents] [Description]

Path 2:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
196. public static FileOutputStream getOutput() //dbutils.java
        ...
198. FileOutputStream file = new FileOutputStream
(Phpbb2MvnforumConfig.EXPORT_XML);
```

[Table of Contents] [Description]

Path 3:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
72. public void processAdd(GenericRequest request) //membergroupwebhandler.java
        ...
108. BufferedReader reader = new BufferedReader(stringReader);
```

[Table of Contents] [Description]

Path 4:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
45. public static String filter(String input) //htmlnewlinefilter.java
        ...
50. BufferedReader reader = new BufferedReader(stringReader);
```

[Table of Contents] [Description]

Path 5:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
        ...
463. FileInputStream fis = new FileInputStream(fileName);
```

[Table of Contents] [Description]

Path 6:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
65. public class DiskFileItem //diskfileitem.java
      |
      ...
      148. private transient DeferredFileOutputStream dfos;
```

[Table of Contents] [Description]

Path 7:

Query Name - **DoS_by_Unreleased_Resources**

Severity - **Warning**

```
672. private void readObject(ObjectInputStream in) //diskfileitem.java
      |
      ...
      681. FileInputStream input = new FileInputStream(dfosFile);
```

[Table of Contents] [Description]

Hard-Coded Password

CWE ID	259
Description	Hard coded passwords may compromise system security in a way that cannot be easily remedied. It is never a good idea to hardcode a password. Not only does hardcoding a password allow all of the project's developers to view the password, it also makes fixing the problem extremely difficult. Once the code is in production, the password cannot be changed without patching the software. If the account protected by the password is compromised, the owners of the system will be forced to choose between security and availability.
Likelihood of Exploit	Very High
Weakness Ordinality	Primary (Weakness exists independent of other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Common Consequences	Authentication: If hard-coded passwords are used, it is almost certain that malicious users will gain access through the account in question.
Potential Mitigations	Design (for default accounts): Rather than hard code a default username and password for first time logins, utilize a "first login" mode which requires the user to enter a unique strong password. Design (for front-end to back-end connections): Three solutions are possible, although none are complete. The first suggestion involves the use of generated passwords which are changed automatically and must be entered at given time intervals by a system administrator. These passwords will be held in memory and only be valid for the time intervals. Next, the passwords used should be limited

at the back end to only performing actions valid to for the front end, as opposed to having full access. Finally, the messages sent should be tagged and checksummed with time sensitive values so as to prevent replay style attacks.

Demonstrative Examples

The following code uses a hardcoded password to connect to a database:

```
...
  DriverManager.getConnection(url, "scott", "tiger");
...
```

This code will run successfully, but anyone who has access to it will have access to the password. Once the program has shipped, there is no going back from the database user "scott" with a password of "tiger" unless the program is patched. A devious employee with access to this information can use it to break into the system. Even worse, if attackers have access to the bytecode for application, they can use the javap -c command to access the disassembled code, which will contain the values of the passwords used. The result of this operation might look something like the following for the example above: javap -c ConnMngr.class 22: ldc #36; //String jdbc:mysql://ixne.com/rxsql 24: ldc #38; //String scott 26: ldc #17; //String tiger

---

C/C++ Example:

```
int VerifyAdmin(char *password) {
if (strcmp(password, "Mew!")) {
  printf("Incorrect Password!\n");
  return(0)
}
printf("Entering Diagnostic Mode…\n"); return(1);
}
```

Java Example:

```
int VerifyAdmin(String password) {
  if (passwd.Eqauls("Mew!")) {
    return(0)
  } //Diagnostic Mode return(1);
}
```

Every instance of this program can be placed into diagnostic mode with the same password. Even worse is the fact that if this program is distributed as a binary-only

	distribution, it is very difficult to change that password or disable this "functionality."
Context Notes	The use of a hard-coded password has many negative implications -- the most significant of these being a failure of authentication measures under certain circumstances. On many systems, a default administration account exists which is set to a simple default password which is hard-coded into the program or device. This hard-coded password is the same for each device or system of this type and often is not changed or disabled by end users. If a malicious user comes across a device of this kind, it is a simple matter of looking up the default password (which is freely available and public on the internet) and logging in with complete access. In systems which authenticate with a back-end service, hard-coded passwords within closed source or drop-in solution systems require that the back-end service use a password which can be easily discovered. Client-side systems with hard-coded passwords propose even more of a threat, since the extraction of a password from a binary is exceedingly simple.
Node Relationships	ChildOf - Credentials Management (255) Peer - Use of Hard-coded Cryptographic Key (321) Peer - Storing Passwords in a Recoverable Format (257) ChildOf - Weaknesses Examined by SAMATE (630)
Source Taxonomies	7 Pernicious Kingdoms - Password Management: Hard-Coded Password CLASP - Use of hard-coded password
Applicable Platforms	All

Path 1:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
162. public static void main(String[] args) //md5.java
       ...
165. String password = "test";
```

[Table of Contents] [Description]

Path 2:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
43. public class MVNForumGlobal //mvnforumglobal.java
       ...
96. public final static String TEMPLATE_FORGOTPASSWORD_PREFIX =
"forgotpasswordtemplate";
```

[Table of Contents] [Description]

Path 3:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
43. public class MVNForumGlobal //mvnforumglobal.java
    ...
    97. public final static String TEMPLATE_FORGOTPASSWORD_SUBJECT =
    "forgotpasswordtemplate_subject.ftl";
```

[Table of Contents] [Description]

Path 4:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
43. public class MVNForumGlobal //mvnforumglobal.java
    ...
    98. public final static String TEMPLATE_FORGOTPASSWORD_BODY =
    "forgotpasswordtemplate_body.ftl";
```

[Table of Contents] [Description]

Path 5:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
66. public class OnlineUserManager //onlineusermanager.java
    ...
    73. private static final String MVNFORUM_SESSION_PASSWORD =
    "mvnforum.encodedpassword";
```

[Table of Contents] [Description]

Path 6:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
66. public class OnlineUserManager //onlineusermanager.java
    ...
    76. private static final String MVNFORUM_COOKIE_PASSWORD =
    "mvnforum.encodedpassword";
```

[Table of Contents] [Description]

Path 7:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
66. public class OnlineUserManager //onlineusermanager.java
    ...
    80. public static final String PASSWORD_OF_METHOD_REALM = "Realm"; //must not
```

```
    be changed in all cases
```

[Table of Contents] [Description]

Path 8:

Query Name - **Hardcoded_Password**

Severity - **Warning**

```
66. public class OnlineUserManager //onlineusermanager.java
    ...
    81. public static final String PASSWORD_OF_METHOD_CUSTOMIZATION =
    "Remote";//must not be changed in all cases
```

[Table of Contents] [Description]

	Error Message Information Leaks
CWE ID	209
Description	Server messages need to be parsed before being passed on to the user.
Likelihood of Exploit	High
Common Consequences	Confidentiality: Often this will either reveal sensitive information which may be used for a later attack or private information stored in the server.
Potential Mitigations	Design: When an application detects illegal input, error messages should only provide generic feedback, such as "Illegal characters were detected." Messages should provide few, if any, implementation details. Implementation: Any error should be parsed for dangerous revelations. Build: Debugging information should not make its way into a production release. Handle exceptions internally and do not display errors containing potentially sensitive information to a user. Create default error pages if necessary.
Demonstrative Examples	**Java Example:** try { /.../ } catch (Exception e) { System.out.println(e); } Here you are passing much more data than is needed. Another example is passing the SQL exceptions to a WebUser without filtering.
Context Notes	Error messages should not provide attackers with any implementation details when the application detects an illegal action. This includes indicating exactly what is allowable, or exactly what was illegal about the user input. Such detailed information can help an attacker craft

	another attack that now will pass through the validation filters. The first thing an attacker may use -- once an attack has failed -- to stage the next attack is the error information provided by the server. SQL Injection attacks generally probe the server for information in order to stage a successful attack.
Node Relationships	ChildOf - Information Leak (Information Disclosure) (200) ParentOf - Product-Generated Error Message Information Leak (210) ParentOf - Product-External Error Message Information Leak (211)
Source Taxonomies	CLASP - Accidental leaking of sensitive information through error messages
Applicable Platforms	All

Path 1:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
128. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
      |...
      |143. e.printStackTrace();
```

[Table of Contents] [Description]

Path 2:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
      |...
      |177. e.printStackTrace();
```

[Table of Contents] [Description]

Path 3:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
      |...
      |179. e.printStackTrace();
```

[Table of Contents] [Description]

Path 4:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
     |...
     181. e.printStackTrace();
```

[Table of Contents] [Description]

Path 5:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
     |...
     183. e.printStackTrace();
```

[Table of Contents] [Description]

Path 6:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
     |...
     185. e.printStackTrace();
```

[Table of Contents] [Description]

Path 7:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
160. public void convert() //phpbbtomvnforum.java
     |...
     194. e.printStackTrace();
```

[Table of Contents] [Description]

Path 8:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
200. public static void main(String[] args) //phpbbtomvnforum.java
     |...
     226. e.printStackTrace();
```

[Table of Contents] [Description]

Path 9:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
66. public RunningVersion(String phphost, String phpdb, String phpuser, String
phppass, String mvnhost, String mvndb, //runningversion.java
      ...
      85. E.printStackTrace();
```

[Table of Contents] [Description]

Path 10:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
102. public void convert() //runningversion.java
      ...
      130. e.printStackTrace();
```

[Table of Contents] [Description]

Path 11:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
102. public void convert() //runningversion.java
      ...
      139. e.printStackTrace();
```

[Table of Contents] [Description]

Path 12:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
90. public void findByAlternateKey_GroupName(String
groupName) //groupsdaoimpljdbc.java
      ...
      110. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 13:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
61. public void findByPrimaryKey(int memberID) //memberdaoimpl.java
      ...
      82. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 14:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
96. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpl.java
      ...
      118. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 15:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
127. public void findByAlternateKey_MemberName(String
memberName) //memberdaoimpl.java
      ...
      148. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 16:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
177. public void create(int memberID, String memberName, String
memberPassword, //memberdaoimpl.java
      ...
      295. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 17:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
415. public void update(int memberID, // primary key //memberdaoimpl.java
      ...
      515. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 18:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
      105. e.printStackTrace();
```

[Table of Contents] [Description]

Path 19:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
      108. e.printStackTrace();
```

[Table of Contents] [Description]

Path 20:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
60. public String getUserNameFromUserID (int userID) throws ObjectNotFoundException,
DatabaseException //phpbbuserdaoimpl.java
      ...
      84. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 21:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
61. public void findByAlternateKey_RankTitle(String
rankTitle) //rankdaoimpljdbc.java
      ...
      80. sqle.printStackTrace();
```

[Table of Contents] [Description]

Path 22:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
67. public class DBUtils //dbutils.java
      ...
      77. e.printStackTrace();
```

[Table of Contents] [Description]

Path 23:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
123. public static void closeResultSet(ResultSet rs) //dbutils.java
      ...
```

```
128. e.printStackTrace();
```

[Table of Contents] [Description]

Path 24:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
137. public static void closeStatement(Statement statement) //dbutils.java
...
142. e.printStackTrace();
```

[Table of Contents] [Description]

Path 25:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
153. public static void closeConnection(Connection connection) //dbutils.java
...
158. e.printStackTrace();
```

[Table of Contents] [Description]

Path 26:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
167. public static void writeXmlFile(Document doc, String filename) //dbutils.java
...
180. e.printStackTrace();
```

[Table of Contents] [Description]

Path 27:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
167. public static void writeXmlFile(Document doc, String filename) //dbutils.java
...
182. e.printStackTrace();
```

[Table of Contents] [Description]

Path 28:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
125. public static synchronized String getMD5_Base64(String input) //md5.java
```

```
...
134. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 29:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
125. public static synchronized String getMD5_Base64(String input) //md5.java
...
144. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 30:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
162. public static void main(String[] args) //md5.java
...
172. e.printStackTrace();
```

[Table of Contents] [Description]

Path 31:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
162. public static void main(String[] args) //md5.java
...
178. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 32:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
...
76. e.printStackTrace();
```

[Table of Contents] [Description]

Path 33:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
      ...
78. e.printStackTrace();
```

[Table of Contents] [Description]

Path 34:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
45. public class Phpbb2MvnforumConfig //phpbb2mvnforumconfig.java
      ...
101. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 35:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
1051. public static void main(String[] args) //importexportxml.java
      ...
1067. e.printStackTrace();
```

[Table of Contents] [Description]

Path 36:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
1051. public static void main(String[] args) //importexportxml.java
      ...
1070. e.printStackTrace();
```

[Table of Contents] [Description]

Path 37:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
1051. public static void main(String[] args) //importexportxml.java
      ...
1073. e.printStackTrace();
```

[Table of Contents] [Description]

Path 38:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
10. public class XMLUtil //xmlutil.java
      ...
29. e.printStackTrace();
```

[Table of Contents] [Description]

Path 39:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
94. private SAXUnmarshallerHandler createUnmarshallerHandler( Locator
locator ) //unmarshallerimpl.java
      ...
110. e.printStackTrace();
```

[Table of Contents] [Description]

Path 40:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
14. public class XMLUtil //xmlutil.java
      ...
35. e.printStackTrace();
```

[Table of Contents] [Description]

Path 41:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void main(String[] args) //md5.java
      ...
114. e.printStackTrace();
```

[Table of Contents] [Description]

Path 42:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void main(String[] args) //md5.java
      ...
120. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 43:

Query Name - **Verbose_Error_Reporting**

Severity

- **Serious**

```
159. public static synchronized String getMD5_Base64(String input) //md5.java
       ...
168. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 44:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
159. public static synchronized String getMD5_Base64(String input) //md5.java
       ...
177. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 45:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
117. public PHPBBToMvnForum(String phphost, String phpdb, String phpuser, String
phppass, //phpbbtomvnforum.java
       ...
140. E.printStackTrace();
```

[Table of Contents] [Description]

Path 46:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
159. public void convert() throws FileNotFoundException //phpbbtomvnforum.java
       ...
200. e.printStackTrace();
```

[Table of Contents] [Description]

Path 47:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
151. public static String HauDateTimeFromS(long s, int timeZone) //utils.java
       ...
171. e.printStackTrace();
```

[Table of Contents] [Description]

Path 48:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
297. public static void ExportContentToFile(ArrayList lines, String
fileName) //converturl.java
      ...
      307. System.out.print(e);
```

[Table of Contents] [Description]

Path 49:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
311. public static void main(String[] args) throws Exception //converturl.java
      ...
      368. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 50:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
      ...
      134. e, clearIfError, request);
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      ...
      503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
      502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 51:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
62. public class CategoryTreeListenerImpl extends
FtlCategoryTreeListener //categorytreelistenerimpl.java
      ...
      93. e.printStackTrace();
```

[Table of Contents] [Description]

Path 52:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
307. public static int getNumDocs() //attachmentindexer.java
      ...
      323. ioe.printStackTrace();
```

[Table of Contents] [Description]

Path 53:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
306. public static int getNumDocs() //memberindexer.java
      ...
      322. ioe.printStackTrace();
```

[Table of Contents] [Description]

Path 54:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
411. public static int getNumDocs() //postindexer.java
      ...
      427. ioe.printStackTrace();
```

[Table of Contents] [Description]

Path 55:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
91. public void Compress() //jpegencoder.java
      ...
      99. System.out.println("IO Error: " + e.getMessage());
```

[Table of Contents] [Description]

Path 56:

Query Name - **Verbose_Error_Reporting**

Severity

- **Serious**

```
958. void bufferIt(BufferedOutputStream outStream, int code,int
size) //jpegencoder.java
     |...
     |976. System.out.println("IO Error: " + e.getMessage());
```

[Table of Contents] [Description]

Path 57:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
     |...
     |247. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 58:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
206. public static void copyFile(String srcFilename, String destFilename, boolean
overwrite) throws IOException //fileutil.java
     |...
     |254. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 59:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
298. public static void popFile(File srcFile, OutputStream output) throws
IOException //fileutil.java
     |...
     |314. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 60:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
97. public static void createThumbnail(InputStream inputStream, OutputStream
outputStream, int maxWidth, int maxHeight) //imageutil.java
     |...
     |173. e.printStackTrace();
```

[Table of Contents] [Description]

Path 61:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
97. public static void createThumbnail(InputStream inputStream, OutputStream
outputStream, int maxWidth, int maxHeight) //imageutil.java
      ...
179. e.printStackTrace();
```

[Table of Contents] [Description]

Path 62:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
249. public static void writeJpegImage_Sun(Image image, OutputStream outputStream)
throws IOException //imageutil.java
      ...
282. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 63:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
300. public static void writeJpegImage_nonSun(Image image, OutputStream
outputStream) throws IOException //imageutil.java
      ...
308. ex.printStackTrace();
```

[Table of Contents] [Description]

Path 64:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
300. public static void writeJpegImage_nonSun(Image image, OutputStream
outputStream) throws IOException //imageutil.java
      ...
315. ex1.printStackTrace();
```

[Table of Contents] [Description]

Path 65:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
```

```
...
137. e, clearIfError, request);
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 66:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
...
144. handleFatalError(e.getMessage(), e.getException(),
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
502. if ((e==null) || (e.getMessage()==null)) addErrorMessage(message);
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 67:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
      ...
      144. handleFatalError(e.getMessage(), e.getException(),
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      ...
      503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
      502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 68:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
      ...
      148. e, clearIfError, request);
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
      ...
      503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
```

```
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 69:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
...
150. if (e.getException()==null) {
...
152. e, clearIfError, request);
```

▼

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

▼

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

▼

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

▼

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 70:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
...
152. e, clearIfError, request);
```

▼

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
...
503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

▼

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
     |  426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
     |  ...
     |  502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 71:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
106. public static void importXml(File importFile, //importjive.java
     |  ...
     |  155. e.getException(), clearIfError, request);
```

```
497. private static void handleFatalError(String message, Exception
e, //importjive.java
     |  ...
     |  503. else addErrorMessage(message+"<br/>Cause: "+e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
     |  307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
     |  426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
     |  ...
     |  502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 72:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
     |  ...
     |  120. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
     |  ...
```

```
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 73:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
...
123. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 74:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
...
```

```
130. handleFatalError(e.getMessage(), e.getException(),
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
736. addErrorMessage(message);
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 75:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
...
130. handleFatalError(e.getMessage(), e.getException(),
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 76:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
       ...
       134. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
       ...
       738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
       307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
       426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
       ...
       502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 77:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
       ...
       136. if (e.getException()==null) {
       ...
       138. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
       ...
       738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
       307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
       426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
```

```
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 78:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
...
138. e, clearIfError, request);
```

▼

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

▼

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

▼

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

▼

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 79:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
...
141. e.getException().getMessage(), e.getException(),
```

▼

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
736. addErrorMessage(message);
```

▼

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

▼

```
425. public static void addErrorMessage(String message) //importwebhandler.java
     |  426. outputHtmlTableRow(message, "messageTextBoldRed");
```



```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
     |  ...
     |  502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 80:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
105. public static void importXml(File importFile, //importmvnforum.java
     |  ...
     |  141. e.getException().getMessage(), e.getException(),
```



```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
     |  ...
     |  738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```



```
306. protected static void addErrorMessage(String message) //importwebhelper.java
     |  307. ImportWebHandler.addErrorMessage(message);
```



```
425. public static void addErrorMessage(String message) //importwebhandler.java
     |  426. outputHtmlTableRow(message, "messageTextBoldRed");
```



```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
     |  ...
     |  502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 81:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
     |  ...
     |  191. e, clearIfError, request);
```



```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
     |
     |  ...
```

```
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

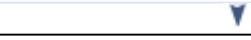
Path 82:

Query Name - **Verbose_Error_Reporting**

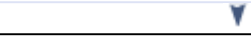
Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
...
194. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```



```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```



```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 83:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
...
```

```
210. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 84:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
...
220. handleFatalError(e.getMessage(), e.getException(),
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
736. addErrorMessage(message);
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 85:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
      220. handleFatalError(e.getMessage(), e.getException(),
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
      ...
      738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
      502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 86:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
      224. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
      ...
      738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
```

```
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 87:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
...
226. if (e.getException()==null) {
...
228. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 88:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
...
228. e, clearIfError, request);
```

```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
...
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```



```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
      502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 89:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
      231. e.getException().getMessage(), e.getException(),
```



```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
      ...
      736. addErrorMessage(message);
```



```
306. protected static void addErrorMessage(String message) //importwebhelper.java
      307. ImportWebHandler.addErrorMessage(message);
```



```
425. public static void addErrorMessage(String message) //importwebhandler.java
      426. outputHtmlTableRow(message, "messageTextBoldRed");
```



```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
      ...
      502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Path 90:

Query Name - **Verbose_Error_Reporting**

Severity - **Serious**

```
175. public static void importZip(File importFile, //importmvnforum.java
      ...
      231. e.getException().getMessage(), e.getException(),
```



```
725. private static void handleFatalError(String message, Exception
e, //importmvnforum.java
      ...
```

```
738. addErrorMessage(message + "<br/>Cause: " + e.getMessage());
```

```
306. protected static void addErrorMessage(String message) //importwebhelper.java
307. ImportWebHandler.addErrorMessage(message);
```

```
425. public static void addErrorMessage(String message) //importwebhandler.java
426. outputHtmlTableRow(message, "messageTextBoldRed");
```

```
495. private static void outputHtmlTableRow(String message, String
cssClass) //importwebhandler.java
...
502. outputHtml.println(" <td class=\""+cssClass+"\">"+message+"</td>");
```

[Table of Contents] [Description]

Resource Injection

CWE ID	99
Description	Allowing user input to control resource identifiers may enable an attacker to access or modify otherwise protected system resources.
Likelihood of Exploit	High
Weakness Ordinality	Primary (Weakness exists independent of other weaknesses)
Causal Nature	Explicit (This is an explicit weakness resulting from behavior of the developer)
Potential Mitigations	Assume all input is malicious. Use an appropriate combination of black lists and white lists to ensure only valid and expected input is processed by the system.
Demonstrative Examples	The following Java code uses input from an HTTP request to create a file name. The programmer has not considered the possibility that an attacker could provide a file name such as "../../tomcat/conf/server.xml", which causes the application to delete one of its own configuration files.

Java Example:

```
String rName = request.getParameter("reportName");
File rFile = new File("/usr/local/apfr/reports/" + rName);
...
rFile.delete();
```

The following code uses input from the command line to determine which file to open and echo back to the user. If the program runs with privileges and malicious users can create soft links to the file, they can use the program to read the first part of any file on the system.

C++ Example:

```
ifstream ifs(argv[0]);
string s;
ifs >> s;
cout << s;
```

The kind of resource the data affects indicates the kind of content that may be dangerous. For example, data containing special characters like period, slash, and backslash, are risky when used in methods that interact with the file system. (Resource injection, when it is related to file system resources, sometimes goes by the name "path manipulation.") Similarly, data that contains URLs and URIs is risky for functions that create remote connections.

Context Notes A resource injection issue occurs when the following two conditions are met: 1. An attacker can specify the identifier used to access a system resource. For example, an attacker might be able to specify part of the name of a file to be opened or a port number to be used. 2. By specifying the resource, the attacker gains a capability that would not otherwise be permitted. For example, the program may give the attacker the ability to overwrite the specified file, run with a configuration controlled by the attacker, or transmit sensitive information to a third-party server. Note: Resource injection that involves resources stored on the filesystem goes by the name path manipulation and is reported in separate category. See the path manipulation description for further details of this vulnerability.

Node Relationships ChildOf - Injection (74)
CanAlsoBe - Path Manipulation (73)
ChildOf - Weaknesses Examined by SAMATE (630)

Source Taxonomies 7 Pernicious Kingdoms - Resource Injection

Applicable Platforms All

Path 1:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
```

```
...
98. return ret;
```

```
584. public void prepareViewLogSystem(GenericRequest
request) //generaladmintaskswebhandler.java
...
594. String shortName = GenericParamUtil.getParameterSafe(request, "filename",
false);
...
610. logFileName = logDir + File.separatorChar + shortName;
...
613. File logFile = new File(logFileName);
```

[Table of Contents] [Description]

Path 2:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
642. public void prepareLogFrame(GenericRequest
request) //generaladmintaskswebhandler.java
...
650. String logFileName = GenericParamUtil.getParameterSafe(request,
"filename", false);
...
665. logFileName = logDir + File.separatorChar + logFileName;
...
668. File logFile = new File(logFileName);
```

[Table of Contents] [Description]

Path 3:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
```

```
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    134. String ret = getParameter(request, param, checkEmpty);
    ...
    141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
    ...
    757. String fileName = ParamUtil.getParameterSafe(request, "filename",
    true);// must check empty here
    ...
    769. File file = new File(MVNForumConfig.getLogDir() + File.separatorChar +
    fileName);
```

[Table of Contents] [Description]

Path 4:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
806. public void deleteLogFile(GenericRequest
request) //generaladmintaskswebhandler.java
    ...
    813. String fileName = GenericParamUtil.getParameterSafe(request, "filename",
    true);
    ...
    825. fileName = logDir + File.separator + fileName;
    ...
    829. if (currentLogFile.equals(new File(fileName))) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
1214. public static void deleteMember(MemberBean memberBean, boolean
deleteSentMessages, GenericRequest request, Locale locale) //memberwebhandler.java
    ...
    1246. bufferPicFile.append(request.getRealPath
    (MVNForumGlobal.UPLOADED_AVATAR_DIR));
    1247. bufferPicFile.append(File.separatorChar).append(memberName).append
    (".jpg");
    1248. String picFile = bufferPicFile.toString();
    ...
    1251. File file = new File(picFile);
```

[Table of Contents] [Description]

Path 6:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
806. public void deleteLogFile(GenericRequest
request) //generaladmintaskswebhandler.java
    ...
    813. String fileName = GenericParamUtil.getParameterSafe(request, "filename",
    true);
    ...
    825. fileName = logDir + File.separator + fileName;
    ...
    833. FileUtil.deleteFile(fileName);
```

```
120. public static void deleteFile(String filename) throws
IOException //fileutil.java
    ...
    122. File file = new File(filename);
```

[Table of Contents] [Description]

Path 7:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      114. String ret = request.getParameter(param);
      ...
      116. ret = ret.trim();
      ...
      122. return ret;
```

```
131. public static String getParameterSafe(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      134. String ret = getParameter(request, param, checkEmpty);
      ...
      141. return ret;
```

```
747. public void downloadLogFile(HttpServletRequest request, HttpServletResponse
response) //generaladmintaskswebhandler.java
      ...
      757. String fileName = ParamUtil.getParameterSafe(request, "filename",
      true);// must check empty here
      ...
      769. File file = new File(MVNForumConfig.getLogDir() + File.separatorChar +
      fileName);
      ...
      791. FileUtil.popFile(file, output);
```

```
298. public static void popFile(File srcFile, OutputStream output) throws
IOException //fileutil.java
      ...
      303. input = new BufferedInputStream(new FileInputStream(srcFile), 4096);
```

[Table of Contents] [Description]

Path 8:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
76. public void processEditCSS(GenericRequest request) //csstaskwebhandler.java
      ...
      88. String CSSPath = request.getRealPath(CSSFile);
      ...
      90. FileUtil.writeFile(cssContent, CSSPath, "UTF-8");
```

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
```

```
...
416. file = new File(fileName);
```

[Table of Contents] [Description]

Path 9:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
76. public void processEditCSS(GenericRequest request) //csstaskwebhandler.java
    ...
    88. String CSSPath = request.getRealPath(CSSFile);
    ...
    90. FileUtil.writeFile(cssContent, CSSPath, "UTF-8");
```

▼

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
    ...
    429. FileOutputStream fos = new FileOutputStream(fileName);
```

[Table of Contents] [Description]

Path 10:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
62. public void prepareEditCSS(GenericRequest request) //csstaskwebhandler.java
    ...
    69. String cssFileName = request.getRealPath(MVNForumGlobal.CSS_FULLPATH);
    70. String cssContent = FileUtil.readFile(cssFileName, "UTF-8");
```

▼

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
    ...
    452. file = new File(fileName);
```

[Table of Contents] [Description]

Path 11:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
62. public void prepareEditCSS(GenericRequest request) //csstaskwebhandler.java
    ...
    69. String cssFileName = request.getRealPath(MVNForumGlobal.CSS_FULLPATH);
    70. String cssContent = FileUtil.readFile(cssFileName, "UTF-8");
```

▼

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
```

```
...
463. FileInputStream fis = new FileInputStream(fileName);
```

[Table of Contents] [Description]

Path 12:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
76. public void processEditCSS(GenericRequest request) //csstaskwebhandler.java
    ...
    88. String CSSPath = request.getRealPath(CSSFile);
    89. FileUtil.touch(CSSPath);// make sure we create this file and update the
    timestamp of this file
```

```
577. public static boolean touch(String fileName) //fileutil.java
    ...
    579. File file = new File(fileName);
```

[Table of Contents] [Description]

Path 13:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
95. public void processEditCSS_forRender(GenericRequest
request) //csstaskwebhandler.java
    ...
    101. String cssFileName = request.getRealPath(cssFile);
    102. String cssContent = FileUtil.readFile(cssFileName, "UTF-8");
```

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
    ...
    452. file = new File(fileName);
```

[Table of Contents] [Description]

Path 14:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
95. public void processEditCSS_forRender(GenericRequest
request) //csstaskwebhandler.java
    ...
    101. String cssFileName = request.getRealPath(cssFile);
    102. String cssContent = FileUtil.readFile(cssFileName, "UTF-8");
```

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
```

```
...
463. FileInputStream fis = new FileInputStream(fileName);
```

[Table of Contents] [Description]

Path 15:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
114. public void processRestoreCSS(GenericRequest request) //csstaskwebhandler.java
...
122. String cssBackup = request.getRealPath
(MVNForumGlobal.CSS_BACKUP_FULLPATH);
123. String fileBackup = FileUtil.readFile(cssBackup, "UTF-8");
```

▼

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
...
452. file = new File(fileName);
```

[Table of Contents] [Description]

Path 16:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
114. public void processRestoreCSS(GenericRequest request) //csstaskwebhandler.java
...
122. String cssBackup = request.getRealPath
(MVNForumGlobal.CSS_BACKUP_FULLPATH);
123. String fileBackup = FileUtil.readFile(cssBackup, "UTF-8");
```

▼

```
447. public static String readFile(String fileName, String
srcEncoding) //fileutil.java
...
463. FileInputStream fis = new FileInputStream(fileName);
```

[Table of Contents] [Description]

Path 17:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
114. public void processRestoreCSS(GenericRequest request) //csstaskwebhandler.java
...
124. String cssFileName = request.getRealPath(MVNForumGlobal.CSS_FULLPATH);
125. FileUtil.writeFile(fileBackup, cssFileName, "UTF-8");
```

▼

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
```

```
...
416. file = new File(fileName);
```

[Table of Contents] [Description]

Path 18:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
114. public void processRestoreCSS(GenericRequest request) //csstaskwebhandler.java
...
124. String cssFileName = request.getRealPath(MVNForumGlobal.CSS_FULLPATH);
125. FileUtil.writeFile(fileBackup, cssFileName, "UTF-8");
```

▼

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
...
429. FileOutputStream fos = new FileOutputStream(fileName);
```

[Table of Contents] [Description]

Path 19:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

▼

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

▼

```
1252. public void processEditTemplate(GenericRequest
request) //configurationwebhandler.java
...
1266. String templateName = GenericParamUtil.getParameterSafe(request,
"template", true);
...
1284. String subjectFileName = templateDir + File.separator + templateName +
SUBJECT_SUFFIX;
...
1292. subjectFileName += EXTENSION;
...
1424. FileUtil.writeFile(subject, subjectFileName, "UTF-8");
```

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
        ...
416.    file = new File(fileName);
```

[Table of Contents] [Description]

Path 20:

Query Name - **Files_Manipulation**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
71.     String ret = request.getParameter(param);
        ...
73.     ret = ret.trim();
        ...
79.     return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
        ...
91.     String ret = getParameter(request, param, checkEmpty);
        ...
98.     return ret;
```

```
1252. public void processEditTemplate(GenericRequest
request) //configurationwebhandler.java
        ...
1266.   String templateName = GenericParamUtil.getParameterSafe(request,
        "template", true);
        ...
1284.   String subjectFileName = templateDir + File.separator + templateName +
        SUBJECT_SUFFIX;
        ...
1292.   subjectFileName += EXTENSION;
        ...
1424.   FileUtil.writeFile(subject, subjectFileName, "UTF-8");
```

```
411. public static void writeFile(String content, String fileName, String
destEncoding) //fileutil.java
        ...
429.    FileOutputStream fos = new FileOutputStream(fileName);
```

[Table of Contents] [Description]

	Cross-Site Request Forgery (CSRF)
CWE ID	352
Description	The web product does not, or can not, sufficiently verify whether a well-formed, valid, consistent request was

	intentionally provided by the user who submitted the request. Note: CSRF is multi-channel: 1. Attacker-to-victim (injection; external or internal channel) 2. Victim-to-server (activation; internal channel)
Alternate Terms	Session Riding Cross Site Reference Forgery XSRF
Observed Examples	CVE-2004-1703 - CVE-2004-1995 - CVE-2004-1967 - CVE-2004-1842 - CVE-2005-1947 - CVE-2005-2059 - CVE-2005-1674 - CSRF
Context Notes	Could be resultant from XSS, although XSS is not necessarily required.
References	Peter W. "Cross-Site Request Forgeries (Re: The Dangers of Allowing Users to Post Images)". Bugtraq. <http://marc.info/?l=bugtraq&m=992631359118 84&w=2>. Robert Auger. "CSRF - The Cross-Site Request Forgery (CSRF/XSRF) FAQ". <http://www.cgisecurity.com/articles/csrf-faq.shtml>.
Node Relationships	ChildOf - Insufficient Verification of Data (345) ChildOf - Weaknesses in OWASP Top Ten (629) ChildOf - Weaknesses Used by NVD (635)
Source Taxonomies	PLOVER - Cross-Site Request Forgery (CSRF)
Applicable Platforms	All

Path 1:

Query Name - **XSRF**

Severity - **Serious**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      120. String[] addList = request.getParameterValues("add");
      ...
      122. int perm = Integer.parseInt(addList[i]);
      ...
      124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
```

```
112. statement.setInt(2, permission);
...
114. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 2:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
    ...
    79. String groupName = GenericParamUtil.getParameterSafe(request, "GroupName",
    true);
    ...
    83. DAOFactory.getGroupsDAO().create(""/*groupOwnerName*/, groupName,
    groupDesc,
```

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
    ...
    166. statement.setString(3, groupName);
    167. statement.setString(4, groupDesc);
    168. statement.setInt(5, groupOption);
    169. statement.setTimestamp(6, groupCreationDate);
    170. statement.setTimestamp(7, groupModifiedDate);
    ...
    172. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 3:

Query Name - **XSRF**

Severity - **Serious**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
```

```
...
123. String[] addList = request.getParameterValues("add");
...
125. int perm = Integer.parseInt(addList[i]);
...
127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
122. statement.setInt(2, permission);
...
124. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 4:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java
...
77. String rankTitle= GenericParamUtil.getParameterSafe(request, "RankTitle",
true);
...
85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```

```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java
...
159. statement.setString(3, rankTitle);
160. statement.setString(4, rankImage);
161. statement.setInt(5, rankType);
162. statement.setInt(6, rankOption);
...
164. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
511. public void processDelete(GenericRequest request) //attachmentwebhandler.java
      ...
      523. int attachID = GenericParamUtil.getParameterInt(request, "attach");
      ...
      555. DAOFactory.getAttachmentDAO().delete(attachID);
```

```
164. public void delete(int attachID) //attachmentdaoimpljdbc.java
      ...
      176. statement.setInt(1, attachID);
      177. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 6:

Query Name - **XSRF**

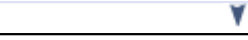
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
```

```
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
597. public void downloadAttachment(GenericRequest request, HttpServletResponse
response) //attachmentwebhandler.java
...
605. int attachID = GenericParamUtil.getParameterInt(request, "attach");
...
646. DAOFactory.getAttachmentDAO().increaseDownloadCount(attachID);
```



```
614. public void increaseDownloadCount(int attachID) //attachmentdaoimpljdbc.java
...
623. statement.setInt(1, attachID);
624. if (statement.executeUpdate() != 1) {
```

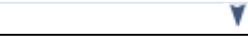
[Table of Contents] [Description]

Path 7:

Query Name - **XSRF**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
...
59. String ret = request.getParameter(param);
...
61. return ret.trim();
```



```
397. public void processEdit(GenericRequest request) //attachmentwebhandler.java
...
433. String newDesc = GenericParamUtil.getParameter(request, "newdesc");
434. DAOFactory.getAttachmentDAO().updateAttachDesc(attachID, newDesc);
```

```
639. public void updateAttachDesc(int attachID, String
attachDesc) //attachmentdaoimpljdbc.java
...
650. statement.setCharacterStream(1, new StringReader(attachDesc),
attachDesc.length());
...
654. statement.setInt(2, attachID);
655. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 8:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
```

```
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
150. public void processDelete(GenericRequest request) //categorywebhandler.java
      ...
      166. int categoryID = GenericParamUtil.getParameterInt(request, "category");
      ...
      186. DAOFactory.getCategoryDAO().delete(categoryID);
```

```
205. public void delete(int categoryID) //categorydaoimpljdbc.java
      ...
      217. statement.setInt(1, categoryID);
      218. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 9:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
216. public void processUpdate(GenericRequest request, GenericResponse
```

```
response) //categorywebhandler.java
    ...
    231. int categoryID = GenericParamUtil.getParameterInt(request, "CategoryID");
    ...
    242. DAOFactory.getCategoryDAO().update(categoryID, // primary key
```

```
236. public void update(int categoryID, // primary key //categorydaoimpljdbc.java
    ...
    275. statement.setInt(7, categoryID);
    ...
    277. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 10:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
253. public void processUpdateCategoryOrder(GenericRequest
request) //categorywebhandler.java
    ...
    266. int categoryID = GenericParamUtil.getParameterInt(request, "category");
    ...
    270. DAOFactory.getCategoryDAO().decreaseCategoryOrder(categoryID, now);
```

```
388. public void decreaseCategoryOrder(int categoryID, Timestamp
categoryModifiedDate) //categorydaoimpljdbc.java
    ...
    398. statement.setInt(2, categoryID);
    399. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 11:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
253. public void processUpdateCategoryOrder(GenericRequest
request) //categorywebhandler.java
      ...
      266. int categoryID = GenericParamUtil.getParameterInt(request, "category");
      ...
      272. DAOFactory.getCategoryDAO().increaseCategoryOrder(categoryID, now);
```

```
417. public void increaseCategoryOrder(int categoryID, Timestamp
categoryModifiedDate) //categorydaoimpljdbc.java
      ...
      427. statement.setInt(2, categoryID);
      428. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 12:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
```

```
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java

...
304. forumID = GenericParamUtil.getParameterInt(request, "forum");
...
454. DAOFactory.getFavoriteThreadDAO().create(memberID, threadID, forumID,
```

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java

...
159. statement.setInt(3, forumID);
160. statement.setTimestamp(4, favoriteCreationDate);
161. statement.setInt(5, favoriteType);
162. statement.setInt(6, favoriteOption);
163. statement.setInt(7, favoriteStatus);
...
165. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 13:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java

...
727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
420. user = getOnlineUser(sessionID);
...
465. return user;
```

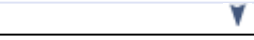
```
114. public void processDelete(GenericRequest
request) //favoritethreadwebhandler.java

...
118. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
```

```
123. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
114. public void processDelete(GenericRequest
request) //favoritethreadwebhandler.java
...
123. int memberID = onlineUser.getMemberID();
...
126. DAOFactory.getFavoriteThreadDAO().delete(memberID, threadID);
```



```
178. public void delete(int memberID, int threadID) //favoritethreaddaoimpljdbc.java
...
190. statement.setInt(1, memberID);
191. statement.setInt(2, threadID);
192. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

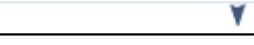
Path 14:

Query Name - **XSRF**

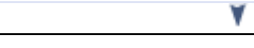
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
...
204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
...
222. DAOFactory.getFavoriteThreadDAO().delete_inForum(forumID);
```



```
230. public void delete_inForum(int forumID) //favoritethreaddaoimpljdbc.java
...
242. statement.setInt(1, forumID);
```

```
...
244. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 15:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
```

```
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
552. public void processMoveThread(GenericRequest request) //threadwebhandler.java
```

```
...
567. int threadID = GenericParamUtil.getParameterInt(request, "thread");
...
599. DAOFactory.getFavoriteThreadDAO().update_ForumID_inThread(threadID,
destForumID);
```

```
282. public void update_ForumID_inThread(int threadID, int
forumID) //favoritethreaddaoimpljdbc.java
```

```
...
306. statement.setInt(2, threadID);
...
308. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 16:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
```

```
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
...
204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
...
230. DAOFactory.getForumDAO().delete(forumID);
```

```
257. public void delete(int forumID) //forumdaoimpljdbc.java
...
269. statement.setInt(1, forumID);
270. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 17:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
...
304. forumID = GenericParamUtil.getParameterInt(request, "forum");
...
431. StatisticsUtil.updateForumStatistics(forumID);
```

```
99. public static void updateForumStatistics(int forumID) //statisticsutil.java
        ...
        119. DAOFactory.getForumDAO().updateLastPostMemberName(forumID,
        lastPostMemberName);
```

```
383. public void updateLastPostMemberName(int forumID, // primary
key //forumdaoimpljdbc.java
        ...
        411. statement.setInt(2, forumID);
        ...
        413. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 18:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
        ...
        304. forumID = GenericParamUtil.getParameterInt(request, "forum");
        ...
        431. StatisticsUtil.updateForumStatistics(forumID);
```

```
99. public static void updateForumStatistics(int forumID) //statisticsutil.java
        ...
        123. DAOFactory.getForumDAO().updateLastPostDate(forumID, forumLastPostDate);
```

```
442. public void updateLastPostDate(int forumID, // primary
key //forumdaoimpljdbc.java
        ...
        459. statement.setInt(2, forumID);
```

```
...
461. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 19:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

▼

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
    304. forumID = GenericParamUtil.getParameterInt(request, "forum");
    ...
    431. StatisticsUtil.updateForumStatistics(forumID);
```

▼

```
99. public static void updateForumStatistics(int forumID) //statisticsutil.java
    ...
    102. int forumThreadCount = DAOFactory.getThreadDAO
    ().getNumberOfEnableThreads_inForum(forumID);
```

▼

```
534. public int getNumberOfEnableThreads_inForum(int
forumID) //threaddaoimpljdbc.java
    ...
    537. return getNumberOfBeans_inForum(forumID, true);
```

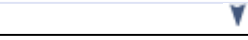
▼

```
546. private int getNumberOfBeans_inForum(int forumID, boolean
enable) //threaddaoimpljdbc.java
    ...
    564. statement.setInt(1, forumID);
    565. resultSet = statement.executeQuery();
    ...
    567. return resultSet.getInt(1);
```

▼

```
534. public int getNumberOfEnableThreads_inForum(int
forumID) //threaddaoimpljdbc.java
      ...
      537. return getNumberOfBeans_inForum(forumID, true);
```

```
99. public static void updateForumStatistics(int forumID) //statisticsutil.java
      ...
      102. int forumThreadCount = DAOFactory.getThreadDAO
      ().getNumberOfEnableThreads_inForum(forumID);
      ...
      110. DAOFactory.getForumDAO().updateStatistics(forumID, forumThreadCount,
      forumPostCount);
```



```
510. public void updateStatistics(int forumID, // primary
key //forumdaoimpljdbc.java
      511. int forumThreadCount, int forumPostCount)
      ...
      524. statement.setInt(1, forumThreadCount);
      525. statement.setInt(2, forumPostCount);
      ...
      528. statement.setInt(3, forumID);
      ...
      530. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 20:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
341. public void processUpdateForumOrder(GenericRequest
request) //forumwebhandler.java
      ...
      351. int forumID = GenericParamUtil.getParameterInt(request, "forum");
      ...
      359. DAOFactory.getForumDAO().decreaseForumOrder(forumID, now);
```

```
800. public void decreaseForumOrder(int forumID, Timestamp
forumModifiedDate) //forumdaoimpljdbc.java
        ...
        810. statement.setInt(2, forumID);
        811. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 21:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
341. public void processUpdateForumOrder(GenericRequest
request) //forumwebhandler.java
        ...
        351. int forumID = GenericParamUtil.getParameterInt(request, "forum");
        ...
        361. DAOFactory.getForumDAO().increaseForumOrder(forumID, now);
```

```
829. public void increaseForumOrder(int forumID, Timestamp
forumModifiedDate) //forumdaoimpljdbc.java
        ...
        839. statement.setInt(2, forumID);
        840. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 22:

Query Name - **XSRF**

Severity - **Serious**

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
        ...
```

```
141. String[] addList = request.getParameterValues("add");
...
143. int perm = Integer.parseInt(addList[i]);
...
145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
...
147. statement.setInt(3, permission);
...
149. if (statement.executeUpdate() != 1) {
```

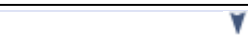
[Table of Contents] [Description]

Path 23:

Query Name - **XSRF**

Severity - **Serious**

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
...
149. String[] removeList = request.getParameterValues("remove");
...
171. int groupID = GenericParamUtil.getParameterInt(request, "group");
```



```
162. public void delete(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
...
176. statement.setInt(3, permission);
177. if (statement.executeUpdate() != 1) {
```

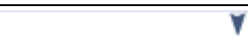
[Table of Contents] [Description]

Path 24:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
```

```
113. return ret;
```

```
128. public void processDelete(GenericRequest request) //groupswebhandler.java
     ...
140. int groupID = GenericParamUtil.getParameterInt(request, "group");
     ...
152. DAOFactory.getGroupForumDAO().delete_inGroup(groupID);
```

```
190. public void delete_inGroup(int groupID) //groupforumdaoimpljdbc.java
     ...
202. statement.setInt(1, groupID);
     ...
204. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 25:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
     ...
71. String ret = request.getParameter(param);
     ...
73. ret = ret.trim();
     ...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
     ...
104. String inputStr = getParameter(request, param, true);
     ...
107. ret = Integer.parseInt(inputStr);
     ...
113. return ret;
```

```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
     ...
204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
     ...
219. DAOFactory.getGroupForumDAO().delete_inForum(forumID);
```

```
215. public void delete_inForum(int forumID) //groupforumdaoimpljdbc.java
     ...
227. statement.setInt(1, forumID);
     ...
229. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 26:

Query Name - **XSRF**

Severity - **Serious**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      120. String[] addList = request.getParameterValues("add");
      ...
      122. int perm = Integer.parseInt(addList[i]);
      ...
      124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      137. statement.setInt(2, permission);
      ...
      139. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 27:

Query Name - **XSRF**

Severity - **Serious**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      128. String[] removeList = request.getParameterValues("remove");
      ...
      130. int perm = Integer.parseInt(removeList[i]);
      ...
      132. DAOFactory.getGroupPermissionDAO().delete(groupID, perm);
```

```
152. public void delete(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      165. statement.setInt(2, permission);
      166. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 28:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
```

```
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
128. public void processDelete(GenericRequest request) //groupswebhandler.java

...
140. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
154. DAOFactory.getGroupPermissionDAO().delete_inGroup(groupID);
```

```
179. public void delete_inGroup(int groupID) //grouppermissiondaoimpljdbc.java

...
191. statement.setInt(1, groupID);
...
193. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 29:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java

...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java

...
79. String groupName = GenericParamUtil.getParameterSafe(request, "GroupName",
true);
...
83. DAOFactory.getGroupsDAO().create(""/*groupOwnerName*/, groupName,
```

```
groupDesc,
```

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
...
189. statement.setString(3, groupName);
...
191. statement.setCharacterStream(4, new StringReader(groupDesc),
groupDesc.length());
...
195. statement.setInt(5, groupOption);
196. statement.setTimestamp(6, groupCreationDate);
197. statement.setTimestamp(7, groupModifiedDate);
...
199. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 30:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
128. public void processDelete(GenericRequest request) //groupswebhandler.java
...
140. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
158. DAOFactory.getGroupsDAO().delete(groupID);
```

```
212. public void delete(int groupID) //groupsdaoimpljdbc.java
...
224. statement.setInt(1, groupID);
225. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 31:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
165. public void processUpdate(GenericRequest request, GenericResponse
response) //groupswebhandler.java
      ...
      182. String groupName = GenericParamUtil.getParameterSafe(request,
      "GroupName", true);
      ...
      186. groupName, groupDesc, now/*groupModifiedDate*/);
```

```
242. public void update(int groupID, // primary key //groupsdaoimpljdbc.java
      243. String groupName, String groupDesc, Timestamp groupModifiedDate)
      ...
      268. statement.setString(1, groupName);
      ...
      270. statement.setCharacterStream(2, new StringReader(groupDesc),
      groupDesc.length());
      ...
      274. statement.setTimestamp(3, groupModifiedDate);
      ...
      277. statement.setInt(4, groupID);
      ...
      279. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 32:

Query Name - **XSRF**

Severity - **Serious**

```
248. protected OnlineUser login(GenericRequest request, GenericResponse
response, //onlineusermanager.java
      ...
      273. String sessionID = request.getSessionId();
      ...
      344. String sessionID = request.getSessionId();
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
```

```
user) //onlineusermanager.java
    ...
    732. OnlineUser onlineUser = getOnlineUser(sessionID);
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    389. return getOnlineUser(request.getServletRequest());
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    615. String memberName = StringUtil.getEmptyStringIfNull
    (authenticator.getRemoteUser(request));
```



```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
    ...
    64. String memberName = request.getRemoteUser();
    65. memberName = StringUtil.getEmptyStringIfNull(memberName);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
    ...
    123. return str;
```



```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
    ...
    65. memberName = StringUtil.getEmptyStringIfNull(memberName);
    ...
    102. return memberName;
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    615. String memberName = StringUtil.getEmptyStringIfNull
    (authenticator.getRemoteUser(request));
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
    ...
    123. return str;
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    615. String memberName = StringUtil.getEmptyStringIfNull
    (authenticator.getRemoteUser(request));
    ...
    618. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      158. statement.setString(1, memberName);
      159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 33:

Query Name - **XSRF**

Severity - **Serious**

```
177. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
      ...
      195. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 34:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
```

```
...
389. return getOnlineUser(request.getServletRequest());
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
...
536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
...
541. int memberID = onlineUser.getMemberID();
...
565. DAOFactory.getMemberDAO().updateEmail(memberID, memberEmail);
```

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
...
509. statement.setInt(2, memberID);
...
511. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 35:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
```

```
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
...
583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
587. int memberID = onlineUser.getMemberID();
```



```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```



```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
...
587. int memberID = onlineUser.getMemberID();
...
624. DAOFactory.getMemberDAO().updatePassword(memberID, // primary key
```



```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
...
563. statement.setInt(2, memberID);
...
573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 36:

Query Name - **XSRF**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
...
59. String ret = request.getParameter(param);
...
61. return ret.trim();
```

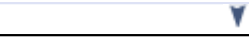


```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
891. String memberName = GenericParamUtil.getParameter(request, "MemberName");
...
```

```
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
944. DAOFactory.getMemberDAO().updateTempPassword(memberID,
currentTempPassword);
```



```
602. public void updateTempPassword(int memberID, // primary
key //memberdaoimpljdbc.java
...
619. statement.setInt(2, memberID);
...
621. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

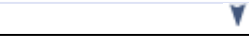
Path 37:

Query Name - **XSRF**

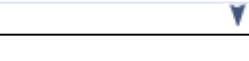
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
```

```
request) //onlineusermanager.java
    ...
508. user = getOnlineUser(sessionID);
    ...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
389. return getOnlineUser(request.getServletRequest());
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
    ...
536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
    ...
541. int memberID = onlineUser.getMemberID();
    ...
563. DAOFactory.getMemberDAO().updateActivateCode(memberID, "");
```

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
    ...
667. statement.setInt(2, memberID);
    ...
669. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 38:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
    ...
```

```
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
735. public void updateMemberAvatar(GenericRequest request) //memberwebhandler.java
...
747. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
752. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
735. public void updateMemberAvatar(GenericRequest request) //memberwebhandler.java
...
752. int memberID = onlineUser.getMemberID();
...
773. DAOFactory.getMemberDAO().updateAvatar(memberID, memberPicture);
```

```
698. public void updateAvatar(int memberID, // primary key //memberdaoimpljdbc.java
...
715. statement.setInt(2, memberID);
...
717. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 39:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
```

```
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
682. public void processUpdateSignature(GenericRequest request, GenericResponse
response) //memberwebhandler.java
...
687. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
693. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
682. public void processUpdateSignature(GenericRequest request, GenericResponse
response) //memberwebhandler.java
...
693. int memberID = onlineUser.getMemberID();
...
701. DAOFactory.getMemberDAO().updateSignature(memberID, // primary key
```

```
746. public void updateSignature(int memberID, // primary
key //memberdaoimpljdbc.java
...
763. statement.setInt(2, memberID);
...
765. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 40:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
475. public void processUpdateMemberTitle(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
      490. String memberTitle = GenericParamUtil.getParameterSafe(request,
      "MemberTitle", false);
      ...
      493. DAOFactory.getMemberDAO().updateTitle(memberID, memberTitle);
```

```
794. public void updateTitle(int memberID, // primary key //memberdaoimpljdbc.java
      795. String memberTitle)
      ...
      808. statement.setString(1, memberTitle);
      ...
      811. statement.setInt(2, memberID);
      ...
      813. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 41:

Query Name - **XSRF**

Severity - **Serious**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      538. memberName = Encoder.decodeURL(cookie.getValue());
```

```
160. public static String decodeURL(String input) //encoder.java
      ...
```

```
162. return URLDecoder.decode(input, "UTF-8");
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
    ...
    548. user = login(request, null, memberName, encodedPassword, true);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    81. String loginName, String password,
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
    137. String key = new String("getMemberIDFromMemberName_" + memberName);
    138. Integer memberIDInterger = (Integer)cache.retrieve(key);
    ...
    144. memberID = memberIDInterger.intValue();
    ...
    154. return memberID;
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
    ...
    132. DAOFactory.getMemberDAO().updateLastLogon(memberID, now,
    request.getRemoteAddr());
```

```
317. public void updateLastLogon(int memberID, Timestamp memberLastLogon, String
lastIP) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    318. memberDAO.updateLastLogon(memberID, memberLastLogon, lastIP);
```

```
842. public void updateLastLogon(int memberID, // primary
key //memberdaoimpljdbc.java
        ...
        860. statement.setInt(3, memberID);
        ...
        862. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 42:

Query Name - **XSRF**

Severity - **Serious**

```
781. public void processListPendingUsers(GenericRequest
request) //memberwebhandler.java
        ...
        793. for (Enumeration enumeration = request.getParameterNames();
        enumeration.hasMoreElements(); ) {
        794. String param = (String) enumeration.nextElement();
        ...
        797. String strMemberID = param.substring(prefix.length());
        ...
        801. memberID = Integer.parseInt(strMemberID);
        ...
        812. DAOFactory.getMemberDAO().updateStatus(memberID,
        MemberBean.MEMBER_STATUS_ENABLE);
```

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
        ...
        2014. statement.setInt(2, memberID);
        ...
        2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 43:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        392. String sessionID = request.getSessionId();
        ...
        420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
        ...
        727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
```

```
...
420. user = getOnlineUser(sessionID);
...
465. return user;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
...
250. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
253. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
...
253. int memberID = onlineUser.getMemberID();
...
430. StatisticsUtil.updateMemberStatistics(memberID);
```

```
127. public static void updateMemberStatistics(int memberID) //statisticsutil.java
...
132. DAOFactory.getMemberDAO().updatePostCount(memberID, memberPostCount);
```

```
339. public void updatePostCount(int memberID, int memberPostCount) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
340. memberDAO.updatePostCount(memberID, memberPostCount);
```

```
2032. public void updatePostCount(int memberID, // primary
key //memberdaoimpljdbc.java
...
2049. statement.setInt(2, memberID);
...
2051. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 44:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
```

```
79. return ret;
```

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
...
1140. String memberName = GenericParamUtil.getParameter(request, "member",
false);
...
1151. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
...
1151. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1159. DAOFactory.getMemberDAO().increaseViewCount(memberID);
```

```
278. public void increaseViewCount(int memberID) throws DatabaseException,
ObjectNotFoundException //memberdaoexternaluserdatabaseabstract.java
279. memberDAO.increaseViewCount(memberID);
```

```
2070. public void increaseViewCount(int memberID) //memberdaoimpljdbc.java
...
2079. statement.setInt(1, memberID);
2080. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **XSRF**

Severity - **Serious**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
...
843. String[] checkList = request.getParameterValues("selectedmember");
...
851. int memberID = Integer.parseInt(checkList[i]);
...
860. DAOFactory.getMemberDAO().updateMember_expireDate(memberID,
memberExpireDate);
```

```
2237. public void updateMember_expireDate(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      2254. statement.setInt(2, memberID);
      ...
      2256. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 46:

Query Name - **XSRF**

Severity - **Serious**

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
      ...
      250. String[] addList = request.getParameterValues("add");
      ...
      252. int perm = Integer.parseInt(addList[i]);
      ...
      254. DAOFactory.getMemberForumDAO().create(memberID, forumID, perm);
```

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
      149. statement.setInt(3, permission);
      ...
      151. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 47:

Query Name - **XSRF**

Severity - **Serious**

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
      ...
      258. String[] removeList = request.getParameterValues("remove");
      ...
      260. int perm = Integer.parseInt(removeList[i]);
      ...
      262. DAOFactory.getMemberForumDAO().delete(memberID, forumID, perm);
```

```
164. public void delete(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
      178. statement.setInt(3, permission);
      179. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 48:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
    ...
    204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
    ...
    220. DAOFactory.getMemberForumDAO().delete_inForum(forumID);
```

```
216. public void delete_inForum(int forumID) //memberforumdaoimpljdbc.java
    ...
    228. statement.setInt(1, forumID);
    ...
    230. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 49:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
```

```
113. return ret;
```

```
130. public void processDelete(GenericRequest request) //membergroupwebhandler.java
     ...
138. int groupID = GenericParamUtil.getParameterInt(request, "group");
     ...
141. DAOFactory.getMemberGroupDAO().delete(groupID, memberID);
```

```
170. public void delete(int groupID, int memberID) //membergroupdaoimpljdbc.java
     ...
182. statement.setInt(1, groupID);
183. statement.setInt(2, memberID);
184. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
     ...
71. String ret = request.getParameter(param);
     ...
73. ret = ret.trim();
     ...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
     ...
104. String inputStr = getParameter(request, param, true);
     ...
107. ret = Integer.parseInt(inputStr);
     ...
113. return ret;
```

```
128. public void processDelete(GenericRequest request) //groupswebhandler.java
     ...
140. int groupID = GenericParamUtil.getParameterInt(request, "group");
     ...
156. DAOFactory.getMemberGroupDAO().delete_inGroup(groupID);
```

```
197. public void delete_inGroup(int groupID) //membergroupdaoimpljdbc.java
     ...
209. statement.setInt(1, groupID);
     ...
211. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 51:

Query Name - **XSRF**

Severity - **Serious**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
    ...
    123. String[] addList = request.getParameterValues("add");
    ...
    125. int perm = Integer.parseInt(addList[i]);
    ...
    127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
    ...
    139. statement.setInt(2, permission);
    ...
    141. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 52:

Query Name - **XSRF**

Severity - **Serious**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
    ...
    131. String[] removeList = request.getParameterValues("remove");
    ...
    133. int perm = Integer.parseInt(removeList[i]);
    ...
    138. DAOFactory.getMemberPermissionDAO().delete(memberID, perm);
```

```
154. public void delete(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
    ...
    167. statement.setInt(2, permission);
    168. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 53:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

▼

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
      ...
      463. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      468. int logonMemberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

▼

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
      ...
      468. int logonMemberID = onlineUser.getMemberID();
      ...
      500. DAOFactory.getMessageDAO().updateMessageReadStatus(messageID,
      logonMemberID, MessageBean.MESSAGE_READ_STATUS_READ);
```

▼

```
656. public void updateMessageReadStatus(int messageID, // primary
key //messagedaoimpljdbc.java
      657. int memberID, int messageReadStatus)
      ...
      675. statement.setInt(3, memberID);
      ...
      677. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 54:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      664. String[] deleteList = request.getParameterValues("selectedmessage");
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      643. int logonMemberID = onlineUser.getMemberID();
      ...
      739. DAOFactory.getMessageDAO().updateFolderName(messageID, logonMemberID,
      destinationFolder);
```

```
722. public void updateFolderName(int messageID, // primary
```

```
key //messagedaoimpljdbc.java
      723. int memberID, String folderName)
      ...
      739. statement.setInt(3, memberID);
      ...
      741. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 55:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

▼

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      664. String[] deleteList = request.getParameterValues("selectedmessage");
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

▼

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      643. int logonMemberID = onlineUser.getMemberID();
      ...
      688. PrivateMessageUtil.deleteMessageInDatabase(messageID, logonMemberID);
```

▼

```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
      ...
      74. DAOFactory.getMessageDAO().deleteMessage(messageID, memberID);
```

▼

```
754. public void deleteMessage(int messageID, int
memberID) //messagedaoimpljdbc.java
      ...
      768. statement.setInt(2, memberID);
      769. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 56:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

▼

```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java
      ...
      213. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      221. int memberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

▼

```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java
      ...
      221. int memberID = onlineUser.getMemberID();
      ...
      240. PrivateMessageUtil.deleteMessageFolderInDatabase(folderName, memberID);
```

▼

```
90. public static void deleteMessageFolderInDatabase(String folderName, int
memberID) //privatemessageutil.java
```

```
...
102. DAOFactory.getMessageDAO().deleteMessages_inFolderName_inMember
(folderName, memberID);
```



```
807. public void deleteMessages_inFolderName_inMember(String folderName, int
memberID) //messagedaoimpljdbc.java

...
821. statement.setInt(2, memberID);
...
823. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 57:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java

...
727. return (OnlineUser)userMap.get(sessionID);
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
420. user = getOnlineUser(sessionID);
...
465. return user;
```



```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java

...
213. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
221. int memberID = onlineUser.getMemberID();
```



```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```



```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java

...
221. int memberID = onlineUser.getMemberID();
```

```
...
240. PrivateMessageUtil.deleteMessageFolderInDatabase(folderName, memberID);
```

```
90. public static void deleteMessageFolderInDatabase(String folderName, int
memberID) //privatemessageutil.java

...
105. DAOFactory.getMessageFolderDAO().delete(folderName, memberID);
```

```
163. public void delete(String folderName, int
memberID) //messagefolderdaoimpljdbc.java

...
176. statement.setInt(2, memberID);
177. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 58:

Query Name - **XSRF**

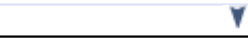
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
389. return getOnlineUser(request.getServletRequest());
```

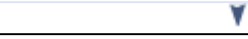
```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java

...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```



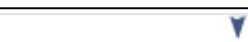
```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java

...
727. return (OnlineUser)userMap.get(sessionID);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java

...
508. user = getOnlineUser(sessionID);
...
630. return user;
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java

...
389. return getOnlineUser(request.getServletRequest());
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
      ...
      255. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      260. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
      ...
      260. int memberID = onlineUser.getMemberID();
      ...
      277. DAOFactory.getMessageFolderDAO().increaseFolderOrder(folderName,
      memberID, now);
```

```
318. public void increaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
      ...
      336. statement.setInt(3, memberID);
      ...
      338. if (statement.executeUpdate() != 1) {
```

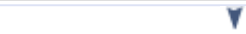
[Table of Contents] [Description]

Path 59:

Query Name - **XSRF**

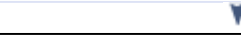
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
```

```
request) //onlineusermanager.java
    ...
508. user = getOnlineUser(sessionID);
    ...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
389. return getOnlineUser(request.getServletRequest());
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
    ...
255. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
260. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
    ...
260. int memberID = onlineUser.getMemberID();
    ...
275. DAOFactory.getMessageFolderDAO().decreaseFolderOrder(folderName,
memberID, now);
```

```
356. public void decreaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
    ...
374. statement.setInt(3, memberID);
    ...
376. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 60:

Query Name - **XSRF**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
114. String ret = request.getParameter(param);
    ...
116. ret = ret.trim();
    ...
122. return ret;
```

```
144. public static int getParameterInt(HttpServletRequest request, String
param) //paramutil.java
        ...
        147. String inputStr = getParameter(request, param, true);
        ...
        150. ret = Integer.parseInt(inputStr);
        ...
        156. return ret;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
        ...
        413. int attachID = ParamUtil.getParameterInt(request, "attach");
        ...
        459. DAOFactory.getPmAttachmentDAO().increaseDownloadCount(attachID);
```

```
283. public void increaseDownloadCount(int
pmAttachID) //pmattachmentdaoimpljdbc.java
        ...
        292. statement.setInt(1, pmAttachID);
        293. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 61:

Query Name - **XSRF**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
        ...
        664. String[] deleteList = request.getParameterValues("selectedmessage");
        ...
        666. int messageID = Integer.parseInt(deleteList[i]);
        ...
        680. PrivateMessageUtil.deleteMessageInDatabase(messageID,
        messageBean.getMemberID());
```

```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
        ...
        77. DAOFactory.getPmAttachMessageDAO().delete_inMessage(messageID);
```

```
193. public void delete_inMessage(int messageID) //pmattachmessagedaoimpljdbc.java
        ...
        205. statement.setInt(1, messageID);
        206. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 62:

Query Name - **XSRF**

Severity

- Serious

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
      ...
      204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
      ...
      226. DAOFactory.getPostDAO().delete_inForum(forumID);
```

```
305. public void delete_inForum(int forumID) //postdaoimpljdbc.java
      ...
      317. statement.setInt(1, forumID);
      318. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 63:

Query Name - **XSRF**

Severity - Serious

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
        ...
        569. int postID = GenericParamUtil.getParameterInt(request, "post");
        ...
        620. DAOFactory.getPostDAO().updateAttachCount(postID, actualAttachCount);
```

```
441. public void updateAttachCount(int postID, // primary key //postdaoimpljdbc.java
        ...
        458. statement.setInt(2, postID);
        ...
        460. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 64:

Query Name - **XSRF**

Severity - **Serious**

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
        ...
        1277. for (Enumeration enumeration = request.getParameterNames();
        enumeration.hasMoreElements(); ) {
        1278. String param = (String) enumeration.nextElement();
        ...
        1281. String strPostID = param.substring(prefix.length());
        1282. int postID = Integer.parseInt(strPostID);
        ...
        1293. DAOFactory.getPostDAO().updateStatus(postID,
        PostBean.POST_STATUS_DEFAULT);
```

```
480. public void updateStatus(int postID, // primary key //postdaoimpljdbc.java
        ...
        497. statement.setInt(2, postID);
        ...
        499. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 65:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
```

```
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
552. public void processMoveThread(GenericRequest request) //threadwebhandler.java
      ...
      567. int threadID = GenericParamUtil.getParameterInt(request, "thread");
      ...
      598. DAOFactory.getPostDAO().update_ForumID_inThread(threadID, destForumID);
```

```
515. public void update_ForumID_inThread(int threadID, int
forumID) //postdaoimpljdbc.java
      ...
      539. statement.setInt(2, threadID);
      ...
      541. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 66:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
      ...
      782. int postID = GenericParamUtil.getParameterInt(request, "post");// dont
      change
      ...
      840. DAOFactory.getPostDAO().increaseEditCount(postID);
```

```
1046. public void increaseEditCount(int postID) //postdaoimpljdbc.java
```

```
...
1055. statement.setInt(1, postID);
1056. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 67:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java
    ...
    77. String rankTitle= GenericParamUtil.getParameterSafe(request, "RankTitle",
    true);
    ...
    85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
    ...
    174. statement.setString(3, rankTitle);
    175. statement.setString(4, rankImage);
    176. statement.setInt(5, rankType);
    177. statement.setInt(6, rankOption);
    ...
    179. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 68:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
```

```
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
147. public void processDelete(GenericRequest request) //rankwebhandler.java
...
156. int rankID = GenericParamUtil.getParameterInt(request, "rank");
...
158. DAOFactory.getRankDAO().delete(rankID);
```

```
257. public void delete(int rankID) //rankdaoimpljdbc.java
...
269. statement.setInt(1, rankID);
270. if (statement.executeUpdate() != 1) {
```

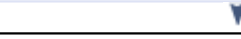
[Table of Contents] [Description]

Path 69:

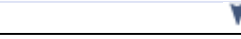
Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
...
204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
```

```
...
227. DAOFactory.getThreadDAO().delete_inForum(forumID);
```

```
244. public void delete_inForum(int forumID) //threaddaoimpljdbc.java
...
256. statement.setInt(1, forumID);
...
258. statement.executeUpdate();
```

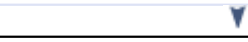
[Table of Contents] [Description]

Path 70:

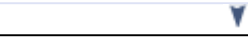
Query Name - **XSRF**

Severity - **Serious**

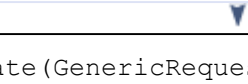
```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



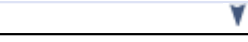
```
128. public static int getParameterInt(GenericRequest request, String param, int
defaultValue) //genericparamutil.java
...
131. String inputStr = getParameter(request, param, false);
...
137. ret = Integer.parseInt(inputStr);
...
143. return ret;
```



```
146. public static int getParameterUnsignedInt(GenericRequest request, String param,
int defaultValue) //genericparamutil.java
...
149. int retValue = getParameterInt(request, param, defaultValue);
...
155. return retValue;
```



```
769. public void processUpdate(GenericRequest request) //postwebhandler.java
...
850. int threadPriority = GenericParamUtil.getParameterUnsignedInt(request,
"ThreadPriority", ThreadBean.THREAD_PRIORITY_NORMAL);
851. DAOFactory.getThreadDAO().updateTopic_Body_Icon(threadID, postTopic,
postBody, threadIcon, threadPriority);
```



```
276. public void updateTopic_Body_Icon(int threadID, // primary
key //threaddaoimpljdbc.java
277. String threadTopic, String threadBody, String threadIcon, int
threadPriority)
...
298. statement.setInt(4, threadPriority);
```

```
...
301. statement.setInt(5, threadID);
...
303. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 71:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
    ...
    1259. threadID = GenericParamUtil.getParameterInt(request, "thread");
    ...
    1316. StatisticsUtil.updateThreadStatistics(threadID);
```

```
66. public static void updateThreadStatistics(int threadID) //statisticsutil.java
    ...
    73. DAOFactory.getThreadDAO().updateThreadAttachCount(threadID,
    attachCountInThread);
```

```
316. public void updateThreadAttachCount(int threadID, //threaddaoimpljdbc.java
    ...
    333. statement.setInt(2, threadID);
    ...
    335. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 72:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
      ...
      1259. threadID = GenericParamUtil.getParameterInt(request, "thread");
      ...
      1316. StatisticsUtil.updateThreadStatistics(threadID);
```

```
66. public static void updateThreadStatistics(int threadID) //statisticsutil.java
      ...
      82. DAOFactory.getThreadDAO().updateLastPostMemberName(threadID,
      lastPostMemberName);
```

```
383. public void updateLastPostMemberName(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
      413. statement.setInt(2, threadID);
      ...
      415. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 73:

Query Name - **XSRF**

Severity - **Serious**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
      ...
      1310. for (Enumeration enumeration = request.getParameterNames();
      enumeration.hasMoreElements(); ) {
      1311. String param = (String) enumeration.nextElement();
      ...
      1314. String strThreadID = param.substring(prefix.length());
      1315. int threadID = Integer.parseInt(strThreadID);
      ...
      1332. DAOFactory.getThreadDAO().updateLastPostDate(threadID, now);
```

```
441. public void updateLastPostDate(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
      458. statement.setInt(2, threadID);
      ...
      460. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 74:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
552. public void processMoveThread(GenericRequest request) //threadwebhandler.java
      ...
      567. int threadID = GenericParamUtil.getParameterInt(request, "thread");
      ...
      597. DAOFactory.getThreadDAO().updateForumID(threadID, destForumID);
```

```
1687. public void updateForumID(int threadID, // primary
key //threaddaoimpljdbc.java
      ...
      1712. statement.setInt(2, threadID);
      ...
      1714. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 75:

Query Name - **XSRF**

Severity - **Serious**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
```

```
...
1310. for (Enumeration enumeration = request.getParameterNames();
enumeration.hasMoreElements(); ) {
1311. String param = (String) enumeration.nextElement();
...
1314. String strThreadID = param.substring(prefix.length());
1315. int threadID = Integer.parseInt(strThreadID);
...
1326. DAOFactory.getThreadDAO().updateThreadStatus(threadID,
ThreadBean.THREAD_STATUS_DEFAULT);
```

```
1730. public void updateThreadStatus(int threadID, // primary
key //threaddaoimpljdbc.java
    ...
    1749. statement.setInt(2, threadID);
    ...
    1751. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 76:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
766. public void processEditThreadType(GenericRequest
request) //threadwebhandler.java
    ...
    780. int threadID = GenericParamUtil.getParameterInt(request, "thread");
    ...
    808. DAOFactory.getThreadDAO().updateThreadType(threadID, threadType);
```

```
1767. public void updateThreadType(int threadID, // primary
key //threaddaoimpljdbc.java
    ...
    1786. statement.setInt(2, threadID);
    ...
```

```
1788. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 77:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
71. String ret = request.getParameter(param);
    ...
73. ret = ret.trim();
    ...
79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
104. String inputStr = getParameter(request, param, true);
    ...
107. ret = Integer.parseInt(inputStr);
    ...
113. return ret;
```

▼

```
1336. public void prepareViewThread(GenericRequest request, GenericResponse
response, String requestURI) //postwebhandler.java
    ...
1346. int threadID = GenericParamUtil.getParameterInt(request, "thread");
    ...
1449. DAOFactory.getThreadDAO().increaseViewCount(threadID);
```

▼

```
1804. public void increaseViewCount(int threadID) //threaddaoimpljdbc.java
    ...
1813. statement.setInt(1, threadID);
1814. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 78:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
71. String ret = request.getParameter(param);
    ...
73. ret = ret.trim();
    ...
79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
        ...
        1259. threadID = GenericParamUtil.getParameterInt(request, "thread");
        ...
        1316. StatisticsUtil.updateThreadStatistics(threadID);
```

```
66. public static void updateThreadStatistics(int threadID) //statisticsutil.java
        ...
        70. DAOFactory.getThreadDAO().updateReplyCount(threadID, threadReplyCount);
```

```
1829. public void updateReplyCount(int threadID, // primary
key //threaddaoimpljdbc.java
        ...
        1850. statement.setInt(2, threadID);
        ...
        1852. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 79:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
415. public void processDelete(GenericRequest request) //watchwebhandler.java
```

```
...
427. int watchID = GenericParamUtil.getParameterInt(request, "watch");
...
439. DAOFactory.getWatchDAO().delete(watchID);
```

```
242. public void delete(int watchID) //watchdaoimpljdbc.java
...
254. statement.setInt(1, watchID);
255. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 80:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
54. public void processDelete_forMember(GenericRequest
request) //watchwebhandler.java
...
58. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
60. DAOFactory.getWatchDAO().delete_inMember(memberID);
```

```
268. public void delete_inMember(int memberID) //watchdaoimpljdbc.java
...
280. statement.setInt(1, memberID);
281. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 81:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
150. public void processDelete(GenericRequest request) //categorywebhandler.java
      ...
      166. int categoryID = GenericParamUtil.getParameterInt(request, "category");
      ...
      184. DAOFactory.getWatchDAO().delete_inCategory(categoryID);
```

```
292. public void delete_inCategory(int categoryID) //watchdaoimpljdbc.java
      ...
      304. statement.setInt(1, categoryID);
      305. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 82:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
195. public void processDelete(GenericRequest request) //forumwebhandler.java
        ...
        204. int forumID = GenericParamUtil.getParameterInt(request, "forum");
        ...
        224. DAOFactory.getWatchDAO().delete_inForum(forumID);
```

```
316. public void delete_inForum(int forumID) //watchdaoimpljdbc.java
        ...
        328. statement.setInt(1, forumID);
        329. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 83:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
465. public void processEdit (GenericRequest request) //watchwebhandler.java
        ...
        480. int watchID = GenericParamUtil.getParameterInt(request, "WatchID");
        ...
        491. DAOFactory.getWatchDAO().updateWatchType(watchID, watchType);
```

```
730. public void updateWatchType(int watchID, int watchType) //watchdaoimpljdbc.java
        ...
        744. statement.setInt(2, watchID);
        ...
        746. if (statement.executeUpdate() < 1) {
```

[Table of Contents] [Description]

Path 84:

Query Name - **XSRF**

Severity - **Serious**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
      ...
      843. String[] checkList = request.getParameterValues("selectedmember");
      ...
      869. int memberID = Integer.parseInt(checkList[i]);
      ...
      872. DAOFactory.getMemberDAO().updateStatus(memberID,
      MemberBean.MEMBER_STATUS_DISABLE);
```

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
      ...
      2014. statement.setInt(2, memberID);
      ...
      2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 85:

Query Name - **XSRF**

Severity - **Serious**

```
338. public void logout(GenericRequest request, GenericResponse
response) //onlineusermanager.java
      ...
      344. String sessionID = request.getSessionId();
      ...
      360. setOnlineUser(sessionID, user);
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
      ...
      732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      615. String memberName = StringUtil.getEmptyStringIfNull
      (authenticator.getRemoteUser(request));
```

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
      ...
      64. String memberName = request.getRemoteUser();
      65. memberName = StringUtil.getEmptyStringIfNull(memberName);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
       123. return str;
```



```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
       ...
       65. memberName = StringUtil.getEmptyStringIfNull(memberName);
       ...
       102. return memberName;
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
       ...
       615. String memberName = StringUtil.getEmptyStringIfNull
       (authenticator.getRemoteUser(request));
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
       ...
       123. return str;
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
       ...
       615. String memberName = StringUtil.getEmptyStringIfNull
       (authenticator.getRemoteUser(request));
       ...
       618. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```



```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
       ...
       158. statement.setString(1, memberName);
       159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 86:

Query Name - **XSRF**

Severity - **Serious**

```
338. public void logout(GenericRequest request, GenericResponse
response) //onlineusermanager.java
       ...
       344. String sessionID = request.getSessionId();
```

[Table of Contents] [Description]

Path 87:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
     ...
589. setOnlineUser(sessionID, user);
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
     ...
732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
430. String memberName = StringUtil.getEmptyStringIfNull(request.getRemoteUser
     ());
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
123. return str;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
430. String memberName = StringUtil.getEmptyStringIfNull(request.getRemoteUser
     ());
     ...
433. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
     ...
158. statement.setString(1, memberName);
159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 88:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

▼

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
    250. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    254. String memberName = onlineUser.getMemberName();
```

▼

```
150. public String getMemberName() //onlineuserimpl.java
    151. return memberName;
```

▼

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
    254. String memberName = onlineUser.getMemberName();
    ...
    320. String lastPostMemberName = memberName;
    ...
    349. threadID = DAOFactory.getThreadDAO().createThread(forumID, memberName,
    lastPostMemberName,
```

▼

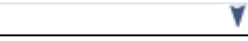
```
198. public int createThread(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
    206. create(forumID, memberName, lastPostMemberName, threadTopic, threadBody,
    threadVoteCount, threadVoteTotalStars, threadCreationDate, threadLastPostDate,
    threadType, threadPriority,threadOption, threadStatus, threadHasPoll,
    threadViewCount, threadReplyCount, threadIcon, threadDuration,
    threadAttachCount);
```

▼

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
```

```
144. DAOFactory.getMemberDAO().findByAlternateKey_MemberName
(lastPostMemberName);
```

```
113. public String findByAlternateKey_MemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    114. return memberDAO.findByAlternateKey_MemberName(memberName);
```



```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    158. statement.setString(1, memberName);
    159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 89:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
```

[Table of Contents] [Description]

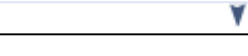
Path 90:

Query Name - **XSRF**

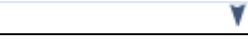
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

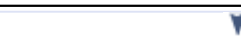
```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```



```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
      ...
      536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      541. int memberID = onlineUser.getMemberID();
```



```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
      ...
      541. int memberID = onlineUser.getMemberID();
      ...
      565. DAOFactory.getMemberDAO().updateEmail(memberID, memberEmail);
```

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
      ...
      509. statement.setInt(2, memberID);
      ...
      511. if (statement.executeUpdate() != 1) {
```

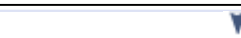
[Table of Contents] [Description]

Path 91:

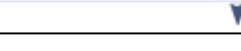
Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
      ...
      583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
```

```
...
587. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
     ...
     587. int memberID = onlineUser.getMemberID();
     ...
     624. DAOFactory.getMemberDAO().updatePassword(memberID, // primary key
```

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
     563. statement.setInt(2, memberID);
     ...
     573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 92:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     392. String sessionID = request.getSessionId();
     ...
     420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     420. user = getOnlineUser(sessionID);
     ...
     465. return user;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
     ...
     536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```



```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
     ...
     541. int memberID = onlineUser.getMemberID();
     ...
     563. DAOFactory.getMemberDAO().updateActivateCode(memberID, "");
```

```
295. public void updateActivateCode(int memberID, String memberActivateCode) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     296. memberDAO.updateActivateCode(memberID, memberActivateCode);
```

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
     667. statement.setInt(2, memberID);
     ...
     669. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 93:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     392. String sessionID = request.getSessionId();
     ...
     420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     420. user = getOnlineUser(sessionID);
     ...
     465. return user;
```

```
735. public void updateMemberAvatar(GenericRequest request) //memberwebhandler.java
     ...
     747. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     752. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
735. public void updateMemberAvatar(GenericRequest request) //memberwebhandler.java
     ...
     752. int memberID = onlineUser.getMemberID();
     ...
     773. DAOFactory.getMemberDAO().updateAvatar(memberID, memberPicture);
```

```
698. public void updateAvatar(int memberID, // primary key //memberdaoimpljdbc.java
     ...
     715. statement.setInt(2, memberID);
     ...
     717. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 94:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     392. String sessionID = request.getSessionId();
     ...
     420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     420. user = getOnlineUser(sessionID);
     ...
     465. return user;
```

```
682. public void processUpdateSignature(GenericRequest request, GenericResponse
response) //memberwebhandler.java
     ...
     687. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     693. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
682. public void processUpdateSignature(GenericRequest request, GenericResponse
response) //memberwebhandler.java
      ...
      693. int memberID = onlineUser.getMemberID();
      ...
      701. DAOFactory.getMemberDAO().updateSignature(memberID, // primary key
```

```
746. public void updateSignature(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      763. statement.setInt(2, memberID);
      ...
      765. if (statement.executeUpdate() != 1) {
```

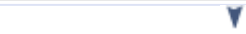
[Table of Contents] [Description]

Path 95:

Query Name - **XSRF**

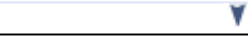
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

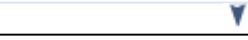


```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

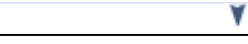
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```



```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
      ...
      95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      148. int memberID = onlineUser.getMemberID();
```

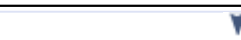


```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```



```
89. public void processAdd(GenericRequest request, GenericResponse
```

```
response) //forumwebhandler.java
    ...
    148. int memberID = onlineUser.getMemberID();
    149. DAOFactory.getMemberForumDAO().create(memberID, forumID,
    MVNForumPermission.PERMISSION_FORUM_ADMIN);
```



```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
    ...
    147. statement.setInt(1, memberID);
    148. statement.setInt(2, forumID);
    149. statement.setInt(3, permission);
    ...
    151. if (statement.executeUpdate() != 1) {
```

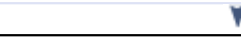
[Table of Contents] [Description]

Path 96:

Query Name - **XSRF**

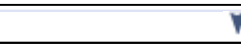
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

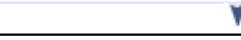


```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

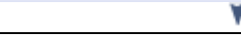
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```



```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
    ...
    463. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    468. int logonMemberID = onlineUser.getMemberID();
```

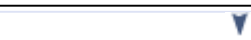


```
146. public int getMemberID() //onlineuserimpl.java
    147. return memberID;
```



```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
```

```
...
468. int logonMemberID = onlineUser.getMemberID();
...
500. DAOFactory.getMessageDAO().updateMessageReadStatus(messageID,
logonMemberID, MessageBean.MESSAGE_READ_STATUS_READ);
```



```
656. public void updateMessageReadStatus(int messageID, // primary
key //messagedaoimpljdbc.java
657. int memberID, int messageReadStatus)
...
675. statement.setInt(3, memberID);
...
677. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 97:

Query Name - **XSRF**

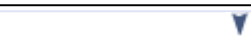
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```

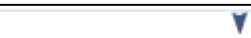
```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
420. user = getOnlineUser(sessionID);
...
465. return user;
```

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
...
664. String[] deleteList = request.getParameterValues("selectedmessage");
```



```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```



```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
...
643. int logonMemberID = onlineUser.getMemberID();
...
739. DAOFactory.getMessageDAO().updateFolderName(messageID, logonMemberID,
```

```
destinationFolder);
```

```
722. public void updateFolderName(int messageID, // primary
key //messagedaoimpljdbc.java
723. int memberID, String folderName)
...
739. statement.setInt(3, memberID);
...
741. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 98:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
420. user = getOnlineUser(sessionID);
...
465. return user;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
...
255. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
260. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
...
260. int memberID = onlineUser.getMemberID();
...
277. DAOFactory.getMessageFolderDAO().increaseFolderOrder(folderName,
```

```
memberID, now);
```

```
318. public void increaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
    ...
    336. statement.setInt(3, memberID);
    ...
    338. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 99:

Query Name - **XSRF**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
    ...
    255. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    260. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
    147. return memberID;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
    ...
    260. int memberID = onlineUser.getMemberID();
    ...
    275. DAOFactory.getMessageFolderDAO().decreaseFolderOrder(folderName,
    memberID, now);
```

```
356. public void decreaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
      ...
      374. statement.setInt(3, memberID);
      ...
      376. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 100:

Query Name - **XSRF**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      664. String[] deleteList = request.getParameterValues("selectedmessage");
      ...
      666. int messageID = Integer.parseInt(deleteList[i]);
      ...
      680. PrivateMessageUtil.deleteMessageInDatabase(messageID,
      messageBean.getMemberID());
```

```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
      ...
      74. DAOFactory.getMessageDAO().deleteMessage(messageID, memberID);
```

```
754. public void deleteMessage(int messageID, int
memberID) //messagedaoimpljdbc.java
      ...
      767. statement.setInt(1, messageID);
      768. statement.setInt(2, memberID);
      769. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 101:

Query Name - **XSRF**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      694. String[] unreadList = request.getParameterValues("selectedmessage");
      ...
      696. int messageID = Integer.parseInt(unreadList[i]);
      ...
      712. DAOFactory.getMessageDAO().updateMessageReadStatus(messageID,
      logonMemberID, MessageBean.MESSAGE_READ_STATUS_DEFAULT);
```

```
656. public void updateMessageReadStatus(int messageID, // primary
key //messagedaoimpljdbc.java
```

```
...
673. statement.setInt(2, messageID);
...
675. statement.setInt(3, memberID);
...
677. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 102:

Query Name - **XSRF**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
...
718. String[] movingList = request.getParameterValues("selectedmessage");
...
723. int messageID = Integer.parseInt(movingList[i]);
...
739. DAOFactory.getMessageDAO().updateFolderName(messageID, logonMemberID,
destinationFolder);
```

```
722. public void updateFolderName(int messageID, // primary
key //messagedaoimpljdbc.java
...
738. statement.setInt(2, messageID);
739. statement.setInt(3, memberID);
...
741. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 103:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      116. int groupID = GenericParamUtil.getParameterInt(request, "group");
      ...
      124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      111. statement.setInt(1, groupID);
      112. statement.setInt(2, permission);
      ...
      114. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 104:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
      ...
      119. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
      ...
      127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
      121. statement.setInt(1, memberID);
      122. statement.setInt(2, permission);
      ...
      124. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 105:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
397. public void processEdit(GenericRequest request) //attachmentwebhandler.java
      ...
      409. int attachID = GenericParamUtil.getParameterInt(request, "attach");
      ...
      434. DAOFactory.getAttachmentDAO().updateAttachDesc(attachID, newDesc);
```

```
639. public void updateAttachDesc(int attachID, String
attachDesc) //attachmentdaoimpljdbc.java
      ...
      654. statement.setInt(2, attachID);
      655. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 106:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
```

```
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
114. public void processDelete(GenericRequest
request) //favoritethreadwebhandler.java
      ...
      124. int threadID = GenericParamUtil.getParameterInt(request, "thread");
      ...
      126. DAOFactory.getFavoriteThreadDAO().delete(memberID, threadID);
```

```
178. public void delete(int memberID, int threadID) //favoritethreaddaoimpljdbc.java
      ...
      191. statement.setInt(2, threadID);
      192. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 107:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
      ...
      134. int groupID = GenericParamUtil.getParameterInt(request, "group");
      ...
      145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
```

```
...
145. statement.setInt(1, groupID);
146. statement.setInt(2, forumID);
147. statement.setInt(3, permission);
...
149. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 108:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
```

```
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

▼

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
```

```
...
134. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
153. DAOFactory.getGroupForumDAO().delete(groupID, forumID, perm);
```

▼

```
162. public void delete(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
```

```
...
174. statement.setInt(1, groupID);
175. statement.setInt(2, forumID);
176. statement.setInt(3, permission);
177. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

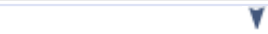
Path 109:

Query Name - **XSRF**

Severity - **Serious**

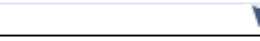
```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java

...
116. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java

...
136. statement.setInt(1, groupID);
137. statement.setInt(2, permission);
...
139. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

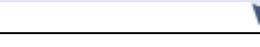
Path 110:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      116. int groupID = GenericParamUtil.getParameterInt(request, "group");
      ...
      132. DAOFactory.getGroupPermissionDAO().delete(groupID, perm);
```

```
152. public void delete(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      164. statement.setInt(1, groupID);
      165. statement.setInt(2, permission);
      166. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 111:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
      ...
      103. String forumOwnerName = GenericParamUtil.getParameterSafe(request,
      "ForumOwnerName", false);
      ...
      131. forumOwnerName = DAOFactory.getMemberDAO().findByAlternateKey_MemberName
      (forumOwnerName);
```

```
113. public String findByAlternateKey_MemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      114. return memberDAO.findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
```

```
...
158. statement.setString(1, memberName);
159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 112:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

▼

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

▼

```
273. public static String getParameterEmail(GenericRequest request, String
param) //genericparamutil.java
...
276. String email = getParameterSafe(request, param, true);
...
278. return email;
```

▼

```
103. public void processAdd(GenericRequest request) //memberwebhandler.java
...
153. String memberEmail = GenericParamUtil.getParameterEmail(request,
"MemberEmail");
...
184. memberEmail, memberEmailVisible, memberNameVisible,
```

▼

```
63. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int memberEmailVisible, int memberNameVisible,
String memberFirstIP, String memberLastIP, int memberViewCount, int memberPostCount,
Timestamp memberCreationDate, Timestamp memberModifiedDate, Timestamp
memberExpireDate, Timestamp memberPasswordExpireDate, Timestamp memberLastLogon, int
memberOption, int memberStatus, String memberActivateCode, String
memberTempPassword, int memberMessageCount, int memberMessageOption, int
memberPostsPerPage, int memberWarnCount, int memberVoteCount, int
memberVoteTotalStars, int memberRewardPoints, String memberTitle, double
memberTimeZone, String memberSignature, String memberAvatar, String memberSkin,
String memberLanguage, String memberFirstname, String memberLastname, int
memberGender, Date memberBirthday, String memberAddress, String memberCity, String
memberState, String memberCountry, String memberPhone, String memberMobile, String
memberFax, String memberCareer, String memberHomepage, String memberYahoo, String
memberAol, String memberIcq, String memberMsn, String memberCoolLink1, String
memberCoolLink2) //memberdaoexternaluserdatabaseabstract.java
```

```
...
66. memberDAO.create(memberName, memberPassword, memberFirstEmail,
memberEmail, memberEmailVisible, memberNameVisible, memberFirstIP,
memberLastIP, memberViewCount, memberPostCount, memberCreationDate,
memberModifiedDate, memberExpireDate, memberPasswordExpireDate,
memberLastLogon, memberOption, memberStatus, memberActivateCode,
memberTempPassword, memberMessageCount, memberMessageOption,
memberPostsPerPage, memberWarnCount, memberVoteCount, memberVoteTotalStars,
memberRewardPoints, memberTitle, memberTimeZone, memberSignature,
memberAvatar, memberSkin, memberLanguage, memberFirstname, memberLastname,
memberGender, memberBirthday, memberAddress, memberCity, memberState,
memberCountry, memberPhone, memberMobile, memberFax, memberCareer,
memberHomepage, memberYahoo, memberAol, memberIcq, memberMsn, memberCoolLink1,
memberCoolLink2);
```

▼

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
    260. String memberEmail, int memberEmailVisible, int memberNameVisible,
    ...
    293. findByAlternateKey_MemberEmail(memberEmail);
```

▼

```
177. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
    ...
    194. statement.setString(1, memberEmail);
    195. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 113:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

▼

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

▼

```
273. public static String getParameterEmail(GenericRequest request, String
param) //genericparamutil.java
    ...
    276. String email = getParameterSafe(request, param, true);
    ...
    278. return email;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
        ...
        544. String memberEmail = GenericParamUtil.getParameterEmail(request,
        "MemberEmail");
        ...
        565. DAOFactory.getMemberDAO().updateEmail(memberID, memberEmail);
```

```
479. public void updateEmail(int memberID, // primary key //memberdaoimpljdbc.java
        480. String memberEmail)
        ...
        506. statement.setString(1, memberEmail);
        ...
        509. statement.setInt(2, memberID);
        ...
        511. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 114:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
        ...
        1010. String memberName = GenericParamUtil.getParameter(request, "member",
        true);
        ...
        1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
        (memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
        ...
        137. String key = new String("getMemberIDFromMemberName_" + memberName);
        138. Integer memberIDInterger = (Integer)cache.retrieve(key);
        ...
        144. memberID = memberIDInterger.intValue();
        ...
        154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
        ...
        1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
        (memberName);
        ...
        1047. DAOFactory.getMemberDAO().updatePassword(memberID, memberPassword,
        passwordExpireDate);
```

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      563. statement.setInt(2, memberID);
      ...
      573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 115:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
      ...
      1010. String memberName = GenericParamUtil.getParameter(request, "member",
      true);
      ...
      1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
      ...
      1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
      ...
      1048. DAOFactory.getMemberDAO().updateTempPassword(memberID, "");// reset the
      temp password
```

```
602. public void updateTempPassword(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      619. statement.setInt(2, memberID);
      ...
      621. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 116:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
      ...
      1059. String memberName = GenericParamUtil.getParameter(request, "MemberName",
      true);
      ...
      1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```

```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
      ...
      1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
      ...
      1081. SendMailUtil.sendActivationCodeEmail(memberID, serverName);
```

```
73. public static void sendActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
      ...
      76. MailMessageStruct mailMessageStruct = getActivationCodeEmail(memberID,
      serverName);
```

```
85. public static MailMessageStruct getActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
      ...
      106. DAOFactory.getMemberDAO().updateActivateCode(memberID, activateCode);
```

```
295. public void updateActivateCode(int memberID, String memberActivateCode) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      |
```

```
296. memberDAO.updateActivateCode(memberID, memberActivateCode);
```

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
...
667. statement.setInt(2, memberID);
...
669. if (statement.executeUpdate() != 1) {
```

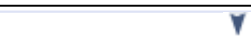
[Table of Contents] [Description]

Path 117:

Query Name - **XSRF**

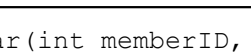
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
519. public void processResetMemberAvatar(GenericRequest
request) //memberwebhandler.java
...
527. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
719. offset = GenericParamUtil.getParameterUnsignedInt(request, "offset");
```



```
698. public void updateAvatar(int memberID, // primary key //memberdaoimpljdbc.java
...
715. statement.setInt(2, memberID);
...
717. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 118:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
500. public void processResetMemberSignature(GenericRequest
request) //memberwebhandler.java
      ...
      508. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
      ...
      510. DAOFactory.getMemberDAO().updateSignature(memberID, "");
```

```
746. public void updateSignature(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      763. statement.setInt(2, memberID);
      ...
      765. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 119:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
```

```
...
113. return ret;
```

```
240. public void processUpdateMemberStatus(GenericRequest
request) //memberwebhandler.java

...
248. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
267. DAOFactory.getMemberDAO().updateStatus(memberID, memberStatus);
```

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java

...
2014. statement.setInt(2, memberID);
...
2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 120:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java

...
119. int retValue = getParameterInt(request, param);
...
125. return retValue;
```

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java

...
841. int expiredDays = GenericParamUtil.getParameterUnsignedInt(request,
"addtime");
...
```

```
848. long offsetDate = expiredDays * DateUtil.DAY;
...
859. Timestamp memberExpireDate = new Timestamp(expireDate.getTime() +
offsetDate);
860. DAOFactory.getMemberDAO().updateMember_expireDate(memberID,
memberExpireDate);
```



```
2237. public void updateMember_expireDate(int memberID, // primary
key //memberdaoimpljdbc.java
2238. Timestamp expireDate)
...
2251. statement.setTimestamp(1, expireDate);
...
2254. statement.setInt(2, memberID);
...
2256. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 121:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
...
243. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
254. DAOFactory.getMemberForumDAO().create(memberID, forumID, perm);
```



```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
...
147. statement.setInt(1, memberID);
148. statement.setInt(2, forumID);
149. statement.setInt(3, permission);
...
151. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 122:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
      ...
      243. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
      ...
      262. DAOFactory.getMemberForumDAO().delete(memberID, forumID, perm);
```

```
164. public void delete(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
      ...
      176. statement.setInt(1, memberID);
      177. statement.setInt(2, forumID);
      178. statement.setInt(3, permission);
      179. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 123:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
        ...
        119. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
        ...
        127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
        ...
        138. statement.setInt(1, memberID);
        139. statement.setInt(2, permission);
        ...
        141. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 124:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
        113. return ret;
```

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
        ...
        119. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
        ...
```

```
138. DAOFactory.getMemberPermissionDAO().delete(memberID, perm);
```

```
154. public void delete(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
166. statement.setInt(1, memberID);
167. statement.setInt(2, permission);
168. if (statement.executeUpdate() != 1) {
```

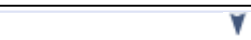
[Table of Contents] [Description]

Path 125:

Query Name - **XSRF**

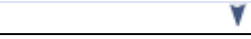
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
451. public void prepareViewMessage(GenericRequest request) //messagewebhandler.java
...
470. int messageID = GenericParamUtil.getParameterInt(request, "message");
...
498. DAOFactory.getMessageDAO().updateMessageReadStatus(messageID,
messageBean.getMemberID(), MessageBean.MESSAGE_READ_STATUS_READ);
```



```
656. public void updateMessageReadStatus(int messageID, // primary
key //messagedaoimpljdbc.java
...
673. statement.setInt(2, messageID);
...
675. statement.setInt(3, memberID);
...
677. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 126:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      719. String destinationFolder = GenericParamUtil.getParameterSafe(request,
      "DestinationFolder", true);
      ...
      739. DAOFactory.getMessageDAO().updateFolderName(messageID, logonMemberID,
      destinationFolder);
```

```
722. public void updateFolderName(int messageID, // primary
key //messagedaoimpljdbc.java
      723. int memberID, String folderName)
      ...
      736. statement.setString(1, folderName);
      ...
      738. statement.setInt(2, messageID);
      739. statement.setInt(3, memberID);
      ...
      741. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 127:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
```

```
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
746. public void processSendMessage(GenericRequest request) //messagewebhandler.java
...
764. int messageID = GenericParamUtil.getParameterInt(request, "message");
...
843. PrivateMessageUtil.deleteMessageInDatabase(messageID, logonMemberID);
```



```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
...
74. DAOFactory.getMessageDAO().deleteMessage(messageID, memberID);
```



```
754. public void deleteMessage(int messageID, int
memberID) //messagedaoimpljdbc.java
...
767. statement.setInt(1, messageID);
768. statement.setInt(2, memberID);
769. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 128:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```



```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java
...
222. String folderName = GenericParamUtil.getParameterSafe(request, "folder",
true);
```

```
...
240. PrivateMessageUtil.deleteMessageFolderInDatabase(folderName, memberID);
```

```
90. public static void deleteMessageFolderInDatabase(String folderName, int
memberID) //privatemessageutil.java
    ...
    102. DAOFactory.getMessageDAO().deleteMessages_inFolderName_inMember
    (folderName, memberID);
```

```
807. public void deleteMessages_inFolderName_inMember(String folderName, int
memberID) //messagedaoimpljdbc.java
    ...
    820. statement.setString(1, folderName);
    821. statement.setInt(2, memberID);
    ...
    823. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 129:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
199. public void processDelete(GenericRequest
request) //messagefolderwebhandler.java
    ...
    222. String folderName = GenericParamUtil.getParameterSafe(request, "folder",
    true);
    ...
    240. PrivateMessageUtil.deleteMessageFolderInDatabase(folderName, memberID);
```

```
90. public static void deleteMessageFolderInDatabase(String folderName, int
memberID) //privatemessageutil.java
    ...
    105. DAOFactory.getMessageFolderDAO().delete(folderName, memberID);
```

```
163. public void delete(String folderName, int
memberID) //messagefolderdaoimpljdbc.java
        ...
        175. statement.setString(1, folderName);
        176. statement.setInt(2, memberID);
        177. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 130:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
        ...
        91. String ret = getParameter(request, param, checkEmpty);
        ...
        98. return ret;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
        ...
        262. String folderName = GenericParamUtil.getParameterSafe(request, "folder",
        true);
        ...
        277. DAOFactory.getMessageFolderDAO().increaseFolderOrder(folderName,
        memberID, now);
```

```
318. public void increaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
        ...
        335. statement.setString(2, folderName);
        336. statement.setInt(3, memberID);
        ...
        338. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 131:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
```

```
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
        ...
        91. String ret = getParameter(request, param, checkEmpty);
        ...
        98. return ret;
```

```
243. public void processUpdateOrder(GenericRequest
request) //messagefolderwebhandler.java
        ...
        262. String folderName = GenericParamUtil.getParameterSafe(request, "folder",
        true);
        ...
        275. DAOFactory.getMessageFolderDAO().decreaseFolderOrder(folderName,
        memberID, now);
```

```
356. public void decreaseFolderOrder(String folderName, int memberID, // primary
key //messagefolderdaoimpljdbc.java
        ...
        373. statement.setString(2, folderName);
        374. statement.setInt(3, memberID);
        ...
        376. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 132:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
        ...
        104. String inputStr = getParameter(request, param, true);
        ...
        107. ret = Integer.parseInt(inputStr);
        ...
```

```
113. return ret;
```



```
746. public void processSendMessage(GenericRequest request) //messagewebhandler.java
    ...
    764. int messageID = GenericParamUtil.getParameterInt(request, "message");
    ...
    843. PrivateMessageUtil.deleteMessageInDatabase(messageID, logonMemberID);
```



```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
    ...
    77. DAOFactory.getPmAttachMessageDAO().delete_inMessage(messageID);
```



```
193. public void delete_inMessage(int messageID) //pmattachmessagedaoimpljdbc.java
    ...
    205. statement.setInt(1, messageID);
    206. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 133:

Query Name - **XSRF**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```



```
1242. public void processModeratePendingPosts(GenericRequest
request) //postwebhandler.java
    ...
    1259. threadID = GenericParamUtil.getParameterInt(request, "thread");
    ...
    1316. StatisticsUtil.updateThreadStatistics(threadID);
```



```
66. public static void updateThreadStatistics(int threadID) //statisticsutil.java
    ...
```

```
86. DAOFactory.getThreadDAO().updateLastPostDate(threadID,
threadLastPostDate);
```

```
441. public void updateLastPostDate(int threadID, // primary
key //threaddaoimpljdbc.java
    ...
    458. statement.setInt(2, threadID);
    ...
    460. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

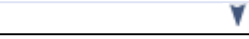
Path 134:

Query Name - **XSRF**

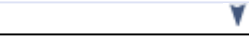
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```



```
664. public void processEditThreadStatus(GenericRequest
request) //threadwebhandler.java
    ...
    678. int threadID = GenericParamUtil.getParameterInt(request, "thread");
    ...
    709. DAOFactory.getThreadDAO().updateThreadStatus(threadID, threadStatus);
```



```
1730. public void updateThreadStatus(int threadID, // primary
key //threaddaoimpljdbc.java
    ...
    1749. statement.setInt(2, threadID);
    ...
    1751. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 135:

Query Name - **XSRF**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      114. String ret = request.getParameter(param);
      ...
      116. ret = ret.trim();
      ...
      122. return ret;
```

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
      ...
      130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
      ...
      140. processLogin(request, response, memberName, memberPassword,
      memberPasswordMD5);
```

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
      147. String memberName, String memberPassword, String memberPasswordMD5)
      ...
      166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
      194. String memberName, String memberPassword, boolean passwordEncoded)
      ...
      211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
      (request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
      ...
      70. String loginName, String password,
      ...
      76. return getAuthenticatedUser(req, null, loginName, password,
      isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
```

```
154. return memberID;
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
    ...
    132. DAOFactory.getMemberDAO().updateLastLogon(memberID, now,
    request.getRemoteAddr());
```

```
317. public void updateLastLogon(int memberID, Timestamp memberLastLogon, String
lastIP) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    318. memberDAO.updateLastLogon(memberID, memberLastLogon, lastIP);
```

```
842. public void updateLastLogon(int memberID, // primary
key //memberdaoimpljdbc.java
    ...
    860. statement.setInt(3, memberID);
    ...
    862. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Web Parameter Tampering

CWE ID	472
Description	A web product does not properly protect assumed-immutable values from modification in hidden form fields, parameters, cookies, or URLs, which lead to modification of critical data.
Alternate Terms	Assumed-Immutable Parameter Tampering
Potential Mitigations	Do not pass hidden fields or cookies (apart from a session cookie) to the server. And, do not use parameters that are not from immediate user input. Assume all input is malicious. Use an appropriate combination of black lists and white lists to ensure only valid and expected input is processed by the system.
Observed Examples	CVE-2002-0108 - Forum product allows spoofed messages of other users via hidden form fields for name and e-mail address. CVE-2000-0253 - Shopping cart allows price modification via hidden form field. CVE-2000-0254 - Shopping cart allows price modification via hidden form field. CVE-2000-0926 - Shopping cart allows price modification via hidden form field. CVE-2000-0101 - Shopping cart allows price modification via

hidden form field.

CVE-2000-0102 - Shopping cart allows price modification via hidden form field.

CVE-2000-0758 - Allows admin access by modifying value of form field.

CVE-2002-1880 - Read messages by modifying message ID parameter.

CVE-2000-1234 - Send email to arbitrary users by modifying email parameter.

CVE-2005-1652 - Authentication bypass by setting a parameter.

CVE-2005-1784 - Product does not check authorization for configuration change admnin script, leading to password theft via modified e-mail address field.

CVE-2005-2314 - Logic error leads to password disclosure.

CVE-2005-1682 - Modification of message number parameter allows attackers to read other people's messages.

Context Notes

This is a primary weakness for many other weaknesses and functional consequences, including XSS, SQL injection, path disclosure, and file inclusion. For example, custom cookies commonly store session data or persistent data across sessions. This kind of session data is normally involved in security related decisions on the server side, such as user authentication and access control. Thus, the cookies might contain sensitive data such as user credentials and privileges. This is a dangerous practice, as it can often lead to improper reliance on the value of the client-provided cookie by the server side application. Without appropriate protection mechanisms, the client can easily tamper with cookies. Reliance on the cookies without detailed validation can lead to problems such as SQL injection. If you use cookie values for security related decisions on the server side, manipulating the cookies might lead to violations of security policies such as authentication bypassing, user impersonation and privilege escalation. In addition, storing sensitive data in the cookie without appropriate protection can also lead to disclosure of sensitive user data, especially data stored in persistent cookies.

Hidden fields should not be trusted as secure parameters. An attacker can intercept and alter hidden fields in a post to the server as easily as user input fields. An attacker can simply parse the HTML for the substring < input type "hidden" or even just "hidden". Hidden field values displayed later in the session, such as on the following page, can open a site up to cross-site scripting attacks.

Node Relationships	This is a technology-specific MAID problem. ChildOf - Modification of Assumed-Immutable Data (MAID) (471) ChildOf - Weaknesses in OWASP Top Ten (629) ParentOf - Unchecked Array Indexing (129)
Source Taxonomies	PLOVER - Web Parameter Tampering
Applicable Platforms	All

Path 1:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      120. String[] addList = request.getParameterValues("add");
      ...
      122. int perm = Integer.parseInt(addList[i]);
      ...
      124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      112. statement.setInt(2, permission);
      ...
      114. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 2:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
```

```
...
134. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
...
131. DAOFactory.getGroupsDAO().findByPrimaryKey(groupID);
```

```
63. public void findByPrimaryKey(int groupID) //groupsdaoimpljdbc.java
...
76. statement.setInt(1, groupID);
77. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 3:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
...
79. String groupName = GenericParamUtil.getParameterSafe(request, "GroupName",
true);
...
83. DAOFactory.getGroupsDAO().create(""/*groupOwnerName*/, groupName,
groupDesc,
```

```
119. public void create(String groupOwnerName, String groupName, String groupDesc,
int groupOption, //groupsdaoimpljdbc.java
...
130. findByAlternateKey_GroupName(groupName);
```

```
90. public void findByAlternateKey_GroupName(String
groupName) //groupsdaoimpljdbc.java
      ...
      103. statement.setString(1, groupName);
      104. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 4:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
      ...
      123. String[] addList = request.getParameterValues("add");
      ...
      125. int perm = Integer.parseInt(addList[i]);
      ...
      127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
      122. statement.setInt(2, permission);
      ...
      124. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 5:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java
      |
```

```
...
77. String rankTitle= GenericParamUtil.getParameterSafe(request, "RankTitle",
true);
...
85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```



```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java

    ...
    130. findByAlternateKey_RankTitle(rankTitle);
```



```
61. public void findByAlternateKey_RankTitle(String
rankTitle) //rankdaoimpljdbc.java

    ...
    74. statement.setString(1, rankTitle);
    75. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 6:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```



```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java

    ...
    76. int rankMinPosts= GenericParamUtil.getParameterInt(request,
    "RankMinPosts");
    ...
    85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```



```
123. public void create(int rankMinPosts, int rankLevel, String rankTitle, String
rankImage, int rankType, int rankOption) //rankdaoimpljdbc.java

    ...
```

```
141. findByAlternateKey_RankMinPosts(rankMinPosts);
```

```
90. public void findByAlternateKey_RankMinPosts(int
rankMinPosts) //rankdaoimpljdbc.java
    ...
    109. statement.setInt(1, rankMinPosts);
    110. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 7:

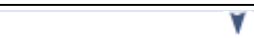
Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
    ...
    569. int postID = GenericParamUtil.getParameterInt(request, "post");
    ...
    614. if (actualAttachCount != DAOFactory.getAttachmentDAO
    ().getNumberOfAttachments_inPost(postID)) {
```



```
341. public int getNumberOfAttachments_inPost(int
postID) //attachmentdaoimpljdbc.java
    ...
    354. statement.setInt(1, postID);
    355. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 8:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
```

```
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
562. public void preparePrintPost(GenericRequest request) //postwebhandler.java
    ...
    569. int postID = GenericParamUtil.getParameterInt(request, "post");
    ...
    609. Collection attachBeans = DAOFactory.getAttachmentDAO
    ().getAttachments_inPost(postID);
```

```
464. public Collection getAttachments_inPost(int
postID) //attachmentdaoimpljdbc.java
    ...
    479. statement.setInt(1, postID);
    480. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 9:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
    ...
    102. int categoryID = GenericParamUtil.getParameterInt(request, "CategoryID");
    ...
    138. int forumID = DAOFactory.getForumDAO().createForum(categoryID,
    forumOwnerName, ""/*lastPostMemberName*/,
```

▼

```
237. public int createForum(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
    ...
    245. create(categoryID, forumOwnerName, lastPostMemberName, forumName,
    forumDesc, forumCreationDate, forumModifiedDate, forumLastPostDate,
```

▼

```
142. public void create(int categoryID, String forumOwnerName, String
lastPostMemberName, //forumdaoimpljdbc.java
    ...
    171. DAOFactory.getCategoryDAO().findByPrimaryKey(categoryID);
```

▼

```
75. public int findByPrimaryKey(int categoryID) //categorydaoimpljdbc.java
    ...
    88. statement.setInt(1, categoryID);
    89. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 10:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

▼

```
126. public void prepareDelete(GenericRequest request) //categorywebhandler.java
    ...
    134. int categoryID = GenericParamUtil.getParameterInt(request, "category");
    ...
```

```
145. CategoryBean categoryBean = DAOFactory.getCategoryDAO().getCategory
(categoryID);
```

```
295. public CategoryBean getCategory(int categoryID) //categorydaoimpljdbc.java
     ...
     308. statement.setInt(1, categoryID);
     309. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 11:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
     479. HttpSession session = request.getSession();
     480. String sessionID = session.getId();
     ...
     508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
     508. user = getOnlineUser(sessionID);
     ...
     630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     389. return getOnlineUser(request.getServletRequest());
```

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
     ...
     65. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     69. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```



```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
     ...
     69. int memberID = onlineUser.getMemberID();
     ...
     72. int currentFavoriteCount = DAOFactory.getFavoriteThreadDAO
     ().getNumberOfFavoriteThreads_inMember(memberID);
```



```
390. public int getNumberOfFavoriteThreads_inMember(int
memberID) //favoritethreaddaoimpljdbc.java
     ...
     403. statement.setInt(1, memberID);
     404. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 12:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
     ...
     71. String ret = request.getParameter(param);
     ...
     73. ret = ret.trim();
     ...
     79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
     ...
     104. String inputStr = getParameter(request, param, true);
     ...
     107. ret = Integer.parseInt(inputStr);
     ...
     113. return ret;
```



```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
     ...
     135. int forumID = GenericParamUtil.getParameterInt(request, "forum");
     ...
     145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```



```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
     ...
     123. DAOFactory.getForumDAO().findByPrimaryKey(forumID);
```

```
74. public void findByPrimaryKey(int forumID) //forumdaoimpljdbc.java
      ...
      87. statement.setInt(1, forumID);
      88. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 13:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

▼

```
163. public void prepareDelete(GenericRequest request) //forumwebhandler.java
      ...
      170. int forumID = GenericParamUtil.getParameterInt(request, "forum");
      ...
      176. forumBean = DAOFactory.getForumDAO().getForum(forumID);
```

▼

```
630. public ForumBean getForum(int forumID) //forumdaoimpljdbc.java
      ...
      643. statement.setInt(1, forumID);
      644. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 14:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
```

```
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
126. public void prepareDelete(GenericRequest request) //categorywebhandler.java
...
134. int categoryID = GenericParamUtil.getParameterInt(request, "category");
...
138. Collection forumsInCategory = DAOFactory.getForumDAO
().getForums_inCategory(categoryID);
```

```
743. public Collection getForums_inCategory(int categoryID) //forumdaoimpljdbc.java
...
758. statement.setInt(1, categoryID);
759. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 15:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
...
141. String[] addList = request.getParameterValues("add");
...
143. int perm = Integer.parseInt(addList[i]);
...
145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```

```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
...
147. statement.setInt(3, permission);
...
149. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 16:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
186. public void prepareAssignGroupToForum(GenericRequest
request) //groupforumwebhandler.java
...
194. int forumID = GenericParamUtil.getParameterInt(request, "forum");
...
205. Collection groupForumBeans = DAOFactory.getGroupForumDAO
().getBeans_inForum(forumID);
```

```
290. public Collection getBeans_inForum(int forumID) //groupforumdaoimpljdbc.java
...
305. statement.setInt(1, forumID);
...
307. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 17:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
162. public void prepareAssignForumToGroup(GenericRequest request, GenericResponse
response) //groupforumwebhandler.java
        ...
        171. int groupID = GenericParamUtil.getParameterInt(request, "group");
        ...
        174. Collection groupForumBeans = DAOFactory.getGroupForumDAO
        ().getBeans_inGroup(groupID);
```

```
330. public Collection getBeans_inGroup(int groupID) //groupforumdaoimpljdbc.java
        ...
        346. statement.setInt(1, groupID);
        ...
        348. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 18:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
        ...
        120. String[] addList = request.getParameterValues("add");
        ...
        122. int perm = Integer.parseInt(addList[i]);
        ...
        124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
        ...
        137. statement.setInt(2, permission);
        ...
        139. if (statement.executeUpdate() != 1) {
```

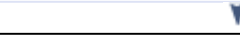
[Table of Contents] [Description]

Path 19:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
        ...
        71. String ret = request.getParameter(param);
        ...
        73. ret = ret.trim();
        ...
        79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
```

```
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
69. public void prepareList(GenericRequest request) //grouppermissionwebhandler.java
...
76. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
80. ArrayList groupPermissionBeans = (ArrayList)
DAOFactory.getGroupPermissionDAO().getBeans_inGroup(groupID);
```



```
214. public Collection getBeans_inGroup(int
groupID) //grouppermissiondaoimpljdbc.java
...
229. statement.setInt(1, groupID);
...
231. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 20:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
...
134. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```



```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
```

```
...
131. DAOFactory.getGroupsDAO().findByPrimaryKey(groupID);
```

```
74. public void findByPrimaryKey(int groupID) //groupsdaoimpljdbc.java
...
87. statement.setInt(1, groupID);
88. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 21:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
...
79. String groupName = GenericParamUtil.getParameterSafe(request, "GroupName",
true);
...
83. DAOFactory.getGroupsDAO().create(""/*groupOwnerName*/, groupName,
groupDesc,
```

```
143. public void create(String groupOwnerName, String
groupName, //groupsdaoimpljdbc.java
...
154. findByAlternateKey_GroupName(groupName);
```

```
102. public void findByAlternateKey_GroupName(String
groupName) //groupsdaoimpljdbc.java
...
119. statement.setString(1, groupName);
120. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 22:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
308. public void prepareSendMail(GenericRequest
request) //generaladmintaskswebhandler.java
      ...
      346. groupID = GenericParamUtil.getParameterInt(request, "group");
      347. to = "Group: " + DAOFactory.getGroupsDAO().getGroup(groupID).getGroupName
      ();
```

```
356. public GroupsBean getGroup(int groupID) //groupsdaoimpljdbc.java
      ...
      369. statement.setInt(1, groupID);
      370. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 23:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
```

```
...
98. return ret;
```

```
66. public void processAdd(GenericRequest request, GenericResponse
response) //groupswebhandler.java
...
79. String groupName = GenericParamUtil.getParameterSafe(request, "GroupName",
true);
...
89. int groupID = DAOFactory.getGroupsDAO().getGroupIDFromGroupName
(groupName);
```

```
510. public int getGroupIDFromGroupName(String groupName) //groupsdaoimpljdbc.java
...
520. statement.setString(1, groupName);
521. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 24:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
589. setOnlineUser(sessionID, user);
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
...
736. userMap.put(sessionID, user);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
589. setOnlineUser(sessionID, user);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
      ...
      732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      426. setOnlineUser(sessionID, user);
```

```
170. public void processAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
      ...
      184. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      373. int receivedMemberID = ((Integer)enumeration.nextElement()).intValue();
      ...
      395. DAOFactory.getMessageStatisticsDAO().create(messageSenderID,
      receivedMemberID, messageCreateDate,
```

```
77. public void create(int fromID, int toID, Timestamp
messageCreationDate, //messagestatisticsdaoimpljdbc.java
      ...
      92. DAOFactory.getMemberDAO().findByPrimaryKey(toID);
```

```
78. public void findByPrimaryKey(int memberID) //memberdaoimpljdbc.java
      ...
      91. statement.setInt(1, memberID);
      92. resultSet = statement.executeQuery();
```

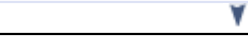
[Table of Contents] [Description]

Path 25:

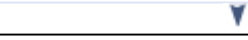
Query Name - **Parameter_Tampering**

Severity - **Serious**

```
248. protected OnlineUser login(GenericRequest request, GenericResponse
response, //onlineusermanager.java
      ...
      273. String sessionID = request.getSessionId();
      ...
      344. String sessionID = request.getSessionId();
```



```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
      ...
      732. OnlineUser onlineUser = getOnlineUser(sessionID);
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
```

```
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
615. String memberName = StringUtil.getEmptyStringIfNull
(authenticator.getRemoteUser(request));
```

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
...
64. String memberName = request.getRemoteUser();
65. memberName = StringUtil.getEmptyStringIfNull(memberName);
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
...
65. memberName = StringUtil.getEmptyStringIfNull(memberName);
...
102. return memberName;
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
615. String memberName = StringUtil.getEmptyStringIfNull
(authenticator.getRemoteUser(request));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
615. String memberName = StringUtil.getEmptyStringIfNull
(authenticator.getRemoteUser(request));
...
618. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
...
158. statement.setString(1, memberName);
159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 26:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
177. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
     ...
195. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 27:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
389. return getOnlineUser(request.getServletRequest());
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
     ...
508. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
508. user = getOnlineUser(sessionID);
     ...
630. return user;
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
389. return getOnlineUser(request.getServletRequest());
```

▼

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
     ...
```

```
583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
587. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
     ...
     587. int memberID = onlineUser.getMemberID();
     ...
     624. DAOFactory.getMemberDAO().updatePassword(memberID, // primary key
```

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
     563. statement.setInt(2, memberID);
     ...
     573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 28:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
     ...
     59. String ret = request.getParameter(param);
     ...
     61. return ret.trim();
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
     891. String memberName = GenericParamUtil.getParameter(request, "MemberName");
     ...
     901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
     (memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
     ...
     137. String key = new String("getMemberIDFromMemberName_" + memberName);
     138. Integer memberIDInterger = (Integer)cache.retrieve(key);
     ...
     144. memberID = memberIDInterger.intValue();
     ...
     154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
     ...
```

```
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
944. DAOFactory.getMemberDAO().updateTempPassword(memberID,
currentTempPassword);
```

```
602. public void updateTempPassword(int memberID, // primary
key //memberdaoimpljdbc.java
...
619. statement.setInt(2, memberID);
...
621. if (statement.executeUpdate() != 1) {
```

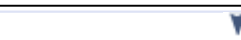
[Table of Contents] [Description]

Path 29:

Query Name - **Parameter_Tampering**

Severity - **Serious**

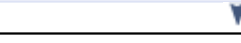
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



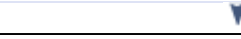
```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
```

```
...
536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
     ...
     541. int memberID = onlineUser.getMemberID();
     ...
     563. DAOFactory.getMemberDAO().updateActivateCode(memberID, "");
```

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
     ...
     667. statement.setInt(2, memberID);
     ...
     669. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 30:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
     538. memberName = Encoder.decodeURL(cookie.getValue());
```

```
160. public static String decodeURL(String input) //encoder.java
     ...
     162. return URLDecoder.decode(input, "UTF-8");
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
     538. memberName = Encoder.decodeURL(cookie.getValue());
     ...
     548. user = login(request, null, memberName, encodedPassword, true);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
     194. String memberName, String memberPassword, boolean passwordEncoded)
     ...
     211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
     (request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
      ...
      70. String loginName, String password,
      ...
      76. return getAuthenticatedUser(req, null, loginName, password,
      isEncodedPassword);
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
      ...
      132. DAOFactory.getMemberDAO().updateLastLogon(memberID, now,
      request.getRemoteAddr());
```



```
317. public void updateLastLogon(int memberID, Timestamp memberLastLogon, String
lastIP) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      318. memberDAO.updateLastLogon(memberID, memberLastLogon, lastIP);
```



```
842. public void updateLastLogon(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      860. statement.setInt(3, memberID);
      ...
      862. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 31:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
```

```
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
...
583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
587. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
...
587. int memberID = onlineUser.getMemberID();
...
613. String currentPassword = DAOFactory.getMemberDAO().getPassword(memberID);
```

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
...
904. statement.setInt(1, memberID);
905. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 32:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      891. String memberName = GenericParamUtil.getParameter(request, "MemberName");
      ...
      901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
      ...
      936. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
```

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
```

[Table of Contents] [Description]

Path 33:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

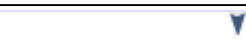
```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
      ...
      536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

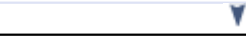
```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
      ...
      541. int memberID = onlineUser.getMemberID();
      ...
      573. SendMailUtil.sendActivationCodeEmail(memberID, serverName);
```

```
73. public static void sendActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
      ...
      76. MailMessageStruct mailMessageStruct = getActivationCodeEmail(memberID,
      serverName);
```

```
85. public static MailMessageStruct getActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
      ...
      90. String activateCode = DAOFactory.getMemberDAO().getActivateCode(memberID);
```



```
137. public String getActivateCode(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      138. return memberDAO.getActivateCode(memberID);
```



```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
      1000. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 34:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
781. public void processListPendingUsers(GenericRequest
request) //memberwebhandler.java
      ...
      793. for (Enumeration enumeration = request.getParameterNames();
      enumeration.hasMoreElements(); ) {
      794. String param = (String) enumeration.nextElement();
      ...
      797. String strMemberID = param.substring(prefix.length());
      ...
      801. memberID = Integer.parseInt(strMemberID);
      802. memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

▼

```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      176. return memberDAO.getMember(memberID);
```

▼

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
      ...
      1220. statement.setInt(1, memberID);
      1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 35:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 36:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
      ...
      893. String memberEmail = GenericParamUtil.getParameter(request,
```

```
"MemberEmail");
...
910. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
(memberEmail);
```

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
    ...
    1446. statement.setString(1, memberEmail);
    1447. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 37:

Query Name - **Parameter_Tampering**

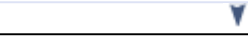
Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

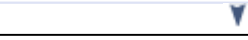
```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    634. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    647. int postsPerPage = onlineUser.getPostsPerPage();
```



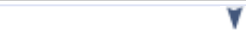
```
323. public int getPostsPerPage() //onlineuserimpl.java
    324. return memberPostsPerPage;
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    647. int postsPerPage = onlineUser.getPostsPerPage();
    ...
```

```
671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
(offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1468. return getBeans_withSortSupport_limit_mysql(offset, rowsToReturn, sort,
order, memberStatus);
```



```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1548. statement.setInt(2, rowsToReturn);
1549. resultSet = statement.executeQuery();
```

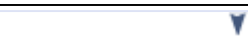
[Table of Contents] [Description]

Path 38:

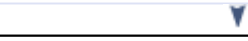
Query Name - **Parameter_Tampering**

Severity - **Serious**

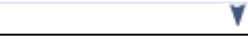
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
727. return (OnlineUser)userMap.get(sessionID);
```

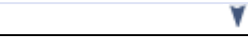


```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
420. user = getOnlineUser(sessionID);
...
465. return user;
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
634. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
647. int postsPerPage = onlineUser.getPostsPerPage();
```

```
323. public int getPostsPerPage() //onlineuserimpl.java
    324. return memberPostsPerPage;
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
        ...
        647. int postsPerPage = onlineUser.getPostsPerPage();
        ...
        671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
        (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
        ...
        1470. return getBeans_withSortSupport_limit_noscroll(offset, rowsToReturn,
        sort, order, memberStatus);
```

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
        ...
        1677. statement.setMaxRows(offset + rowsToReturn);
        1678. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 39:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        392. String sessionID = request.getSessionId();
        ...
        420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
        ...
        727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        420. user = getOnlineUser(sessionID);
        ...
        465. return user;
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
        ...
        634. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
        ...
        647. int postsPerPage = onlineUser.getPostsPerPage();
```

```
323. public int getPostsPerPage() //onlineuserimpl.java
     324. return memberPostsPerPage;
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
     ...
     647. int postsPerPage = onlineUser.getPostsPerPage();
     ...
     671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
     (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
     ...
     1472. return getBeans_withSortSupport_limit_general(offset, rowsToReturn,
     sort, order, memberStatus);
```



```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
     ...
     1812. statement.setMaxRows(offset + rowsToReturn);
     ...
     1814. statement.setFetchSize(Math.min(rowsToReturn, DBUtils.MAX_FETCH_SIZE));
     ...
     1818. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 40:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
781. public void processListPendingUsers(GenericRequest
request) //memberwebhandler.java
     ...
     793. for (Enumeration enumeration = request.getParameterNames();
     enumeration.hasMoreElements(); ) {
     794. String param = (String) enumeration.nextElement();
     ...
     797. String strMemberID = param.substring(prefix.length());
     ...
     801. memberID = Integer.parseInt(strMemberID);
     ...
     812. DAOFactory.getMemberDAO().updateStatus(memberID,
     MemberBean.MEMBER_STATUS_ENABLE);
```



```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
     ...
     2014. statement.setInt(2, memberID);
     ...
     2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 41:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
      250. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      253. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
      253. int memberID = onlineUser.getMemberID();
      ...
      430. StatisticsUtil.updateMemberStatistics(memberID);
```

```
127. public static void updateMemberStatistics(int memberID) //statisticsutil.java
      ...
      132. DAOFactory.getMemberDAO().updatePostCount(memberID, memberPostCount);
```

```
339. public void updatePostCount(int memberID, int memberPostCount) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      340. memberDAO.updatePostCount(memberID, memberPostCount);
```

```
2032. public void updatePostCount(int memberID, // primary
```

```
key //memberdaoimpljdbc.java
    ...
2049. statement.setInt(2, memberID);
    ...
2051. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 42:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
71. String ret = request.getParameter(param);
    ...
73. ret = ret.trim();
    ...
79. return ret;
```

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
    ...
1140. String memberName = GenericParamUtil.getParameter(request, "member",
false);
    ...
1151. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
    ...
144. memberID = memberIDInterger.intValue();
    ...
154. return memberID;
```

```
1137. public void prepareView_forPublic(GenericRequest
request) //memberwebhandler.java
    ...
1151. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
    ...
1159. DAOFactory.getMemberDAO().increaseViewCount(memberID);
```

```
278. public void increaseViewCount(int memberID) throws DatabaseException,
ObjectNotFoundException //memberdaoexternaluserdatabaseabstract.java
    279. memberDAO.increaseViewCount(memberID);
```

```
2070. public void increaseViewCount(int memberID) //memberdaoimpljdbc.java
    ...
```

```
2079. statement.setInt(1, memberID);
2080. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 43:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
392. String sessionID = request.getSessionId();
    ...
420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
420. user = getOnlineUser(sessionID);
    ...
465. return user;
```

▼

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
    ...
705. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
716. int postsPerPage = onlineUser.getPostsPerPage();
```

▼

```
323. public int getPostsPerPage() //onlineuserimpl.java
    324. return memberPostsPerPage;
```

▼

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java
    ...
716. int postsPerPage = onlineUser.getPostsPerPage();
    ...
731. Collection expireMemberBeans = DAOFactory.getMemberDAO
().getMembers_inExpire_limit(expiresoonDate, offset, postsPerPage, sort,
order);
```

▼

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java
    ...
2172. statement.setMaxRows(offset + rowsToReturn);
```

```
2173. statement.setTimestamp(1, expireDate);
...
2175. statement.setFetchSize(Math.min(rowsToReturn, DBUtils.MAX_FETCH_SIZE));
...
2179. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 44:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
```

```
...
843. String[] checkList = request.getParameterValues("selectedmember");
...
851. int memberID = Integer.parseInt(checkList[i]);
...
860. DAOFactory.getMemberDAO().updateMember_expireDate(memberID,
memberExpireDate);
```

```
2237. public void updateMember_expireDate(int memberID, // primary
key //memberdaoimpljdbc.java
```

```
...
2254. statement.setInt(2, memberID);
...
2256. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 45:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
```

```
...
250. String[] addList = request.getParameterValues("add");
...
252. int perm = Integer.parseInt(addList[i]);
...
254. DAOFactory.getMemberForumDAO().create(memberID, forumID, perm);
```

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
```

```
...
149. statement.setInt(3, permission);
...
151. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 46:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
77. public void prepareAssignForumToMember(GenericRequest request, GenericResponse
response) //memberforumwebhandler.java
    ...
    86. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
    ...
    95. Collection memberForumBeans = DAOFactory.getMemberForumDAO
    ().getBeans_inMember(memberID);
```

```
285. public Collection getBeans_inMember(int memberID) //memberforumdaoimpljdbc.java
    ...
    300. statement.setInt(1, memberID);
    301. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 47:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
```

```
...
113. return ret;
```

```
186. public void prepareAssignGroupToForum(GenericRequest
request) //groupforumwebhandler.java
...
194. int forumID = GenericParamUtil.getParameterInt(request, "forum");
...
206. Collection memberForumBeans = DAOFactory.getMemberForumDAO
().getBeans_inForum(forumID);
```

```
324. public Collection getBeans_inForum(int forumID) //memberforumdaoimpljdbc.java
...
341. statement.setInt(1, forumID);
342. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 48:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
148. public void prepareList_inGroup_limit(GenericRequest
request) //membergroupwebhandler.java
...
157. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
172. Collection memberGroupBeans = DAOFactory.getMemberGroupDAO
().getBeans_inGroup(groupID);
```

```
254. public Collection getBeans_inGroup(int groupID) //membergroupdaoimpljdbc.java
...
269. statement.setInt(1, groupID);
270. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 49:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
      ...
      123. String[] addList = request.getParameterValues("add");
      ...
      125. int perm = Integer.parseInt(addList[i]);
      ...
      127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
      ...
      139. statement.setInt(2, permission);
      ...
      141. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 50:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
70. public void prepareListPermission(GenericRequest
request) //memberpermissionwebhandler.java
      ...
      78. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
      ...
      81. ArrayList memberPermissionBeans = (ArrayList)
      DAOFactory.getMemberPermissionDAO().getBeans_inMember(memberID);
```

```
209. public Collection getBeans_inMember(int
memberID) //memberpermissiondaoimpljdbc.java
      ...
      223. statement.setInt(1, memberID);
      224. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 51:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      664. String[] deleteList = request.getParameterValues("selectedmessage");
      ...
      666. int messageID = Integer.parseInt(deleteList[i]);
      ...
      670. messageBean = DAOFactory.getMessageDAO().getMessage(messageID);
```

```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java
      ...
      586. statement.setInt(1, messageID);
      587. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 52:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
```

```
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
118. public void prepareList(GenericRequest request) //messagefolderwebhandler.java
...
129. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
134. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

```
118. public void prepareList(GenericRequest request) //messagefolderwebhandler.java
...
134. int memberID = onlineUser.getMemberID();
...
136. Collection messageFolderBeans = DAOFactory.getMessageFolderDAO
().getMessageFolders_inMember(memberID);
```

```
244. public Collection getMessageFolders_inMember(int
memberID) //messagefolderdaoimpljdbc.java
...
259. statement.setInt(1, memberID);
260. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 53:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
84. public void processAdd(GenericRequest request) //messagefolderwebhandler.java
      ...
      98. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      110. int folderOrder = DAOFactory.getMessageFolderDAO().getMaxFolderOrder
      (onlineUser.getMemberID());
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
84. public void processAdd(GenericRequest request) //messagefolderwebhandler.java
      ...
      110. int folderOrder = DAOFactory.getMessageFolderDAO().getMaxFolderOrder
      (onlineUser.getMemberID());
```

```
284. public int getMaxFolderOrder(int memberID) //messagefolderdaoimpljdbc.java
      ...
      299. statement.setInt(1, memberID);
      300. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 54:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
```

```
79. return ret;
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
    ...
    376. String strMemberName = GenericParamUtil.getParameter(request, "member",
    false);
    ...
    389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (strMemberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    1416. statement.setString(1, memberName);
    1417. resultSet = statement.executeQuery();
    ...
    1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
    ...
    389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (strMemberName);
    ...
    449. fromMessageCount = DAOFactory.getMessageStatisticsDAO
    ().getNumberOfBeans_inFromID(memberID);
```

```
152. public int getNumberOfBeans_inFromID(int
fromID) //messagestatisticsdaoimpljdbc.java
    ...
    165. statement.setInt(1, fromID);
    166. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 55:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
```

```
...
73. ret = ret.trim();
...
79. return ret;
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
...
376. String strMemberName = GenericParamUtil.getParameter(request, "member",
false);
...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(strMemberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(strMemberName);
...
448. toMessageCount = DAOFactory.getMessageStatisticsDAO
().getNumberOfBeans_inToID(memberID);
```

```
179. public int getNumberOfBeans_inToID(int
toID) //messagestatisticsdaoimpljdbc.java
...
192. statement.setInt(1, toID);
193. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 56:

Query Name - **Parameter_Tampering**

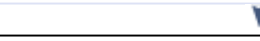
Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
```

```
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

```
144. public static int getParameterInt(HttpServletRequest request, String
param) //paramutil.java

...
147. String inputStr = getParameter(request, param, true);
...
150. ret = Integer.parseInt(inputStr);
...
156. return ret;
```



```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java

...
413. int attachID = ParamUtil.getParameterInt(request, "attach");
...
418. pmAttachBean = DAOFactory.getPmAttachmentDAO().getPmAttachment(attachID);
```

```
235. public PmAttachmentBean getPmAttachment(int
pmAttachID) //pmattachmentdaoimpljdbc.java

...
248. statement.setInt(1, pmAttachID);
249. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

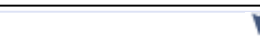
Path 57:

Query Name - **Parameter_Tampering**

Severity - **Serious**

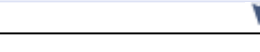
```
620. public void processMessage(GenericRequest request) //messagewebhandler.java

...
664. String[] deleteList = request.getParameterValues("selectedmessage");
...
666. int messageID = Integer.parseInt(deleteList[i]);
...
680. PrivateMessageUtil.deleteMessageInDatabase(messageID,
messageBean.getMemberID());
```



```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java

...
77. DAOFactory.getPmAttachMessageDAO().delete_inMessage(messageID);
```



```
193. public void delete_inMessage(int messageID) //pmattachmessagedaoimpljdbc.java

...
205. statement.setInt(1, messageID);
206. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 58:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
      280. int parentPostID = GenericParamUtil.getParameterInt(request, "parent");
      ...
      416. int postID = DAOFactory.getPostDAO().createPost(parentPostID, forumID,
      threadID,
```

```
224. public int createPost(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
      233. parentPostID, forumID, threadID, memberID, memberName,
      lastEditMemberName,
```

```
110. public void create(int parentPostID, int forumID, int
threadID, //postdaoimpljdbc.java
      ...
      173. findByPrimaryKey(parentPostID);
```

```
75. public void findByPrimaryKey(int postID) //postdaoimpljdbc.java
      ...
      88. statement.setInt(1, postID);
      89. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 59:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
      250. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      253. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
      ...
      253. int memberID = onlineUser.getMemberID();
      ...
      430. StatisticsUtil.updateMemberStatistics(memberID);
```

```
127. public static void updateMemberStatistics(int memberID) //statisticsutil.java
      ...
      130. int memberPostCount = DAOFactory.getPostDAO().getNumberOfPosts_inMember
      (memberID);
```

```
891. public int getNumberOfPosts_inMember(int memberID) //postdaoimpljdbc.java
      ...
      905. statement.setInt(1, memberID);
      ...
      907. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 60:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java
      ...
      77. String rankTitle= GenericParamUtil.getParameterSafe(request, "RankTitle",
      true);
      ...
      85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java
      ...
      145. findByAlternateKey_RankTitle(rankTitle);
```

```
72. public void findByAlternateKey_RankTitle(String
rankTitle) //rankdaoimpljdbc.java
      ...
      89. statement.setString(1, rankTitle);
      90. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 61:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
```

```
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
64. public void processAdd(GenericRequest request, GenericResponse
response) //rankwebhandler.java

...
76. int rankMinPosts= GenericParamUtil.getParameterInt(request,
"RankMinPosts");
...
85. DAOFactory.getRankDAO().create(rankMinPosts, rankLevel, rankTitle,
```

```
137. public void create(int rankMinPosts, int rankLevel, String
rankTitle, //rankdaoimpljdbc.java

...
156. findByAlternateKey_RankMinPosts(rankMinPosts);
```

```
104. public void findByAlternateKey_RankMinPosts(int
rankMinPosts) //rankdaoimpljdbc.java

...
117. statement.setInt(1, rankMinPosts);
118. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 62:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
```

```
...
113. return ret;
```

```
93. public void prepareEdit(GenericRequest request) //rankwebhandler.java
...
102. int rankID = GenericParamUtil.getParameterInt(request, "rank");
...
104. RankBean rankBean = DAOFactory.getRankDAO().getRank(rankID);
```

```
288. public RankBean getRank(int rankID) //rankdaoimpljdbc.java
...
301. statement.setInt(1, rankID);
302. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 63:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
...
83. int threadID = GenericParamUtil.getParameterInt(request, "thread");
...
106. DAOFactory.getFavoriteThreadDAO().create(memberID, threadID, forumID,
```

```
107. public void create(int memberID, int threadID, int
forumID, //favoritethreaddaoimpljdbc.java
...
135. DAOFactory.getThreadDAO().findByPrimaryKey(threadID);
```

```
74. public void findByPrimaryKey(int threadID) //threaddaoimpljdbc.java
...
```

```
87. statement.setInt(1, threadID);
88. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 64:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
1280. public void processModeratePendingThreads(GenericRequest
request) //threadwebhandler.java
      ...
      1310. for (Enumeration enumeration = request.getParameterNames();
      enumeration.hasMoreElements(); ) {
      1311. String param = (String) enumeration.nextElement();
      ...
      1314. String strThreadID = param.substring(prefix.length());
      1315. int threadID = Integer.parseInt(strThreadID);
      ...
      1319. threadBean = DAOFactory.getThreadDAO().getThread(threadID);
```

```
480. public ThreadBean getThread(int threadID) //threaddaoimpljdbc.java
      ...
      493. statement.setInt(1, threadID);
      494. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 65:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
415. public void processDelete(GenericRequest request) //watchwebhandler.java
      ...
      427. int watchID = GenericParamUtil.getParameterInt(request, "watch");
```

```
...
429. WatchBean watchBean = DAOFactory.getWatchDAO().getWatch(watchID);
```

```
406. public WatchBean getWatch(int watchID) //watchdaoimpljdbc.java

...
419. statement.setInt(1, watchID);
420. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 66:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java

...
376. String strMemberName = GenericParamUtil.getParameter(request, "member",
false);
...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(strMemberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java

...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java

...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
```

```
(strMemberName);
...
460. int watchCount = DAOFactory.getWatchDAO().getNumberOfWatches_forMember
(memberID);
```



```
574. public int getNumberOfWatches_forMember(int memberID) //watchdaoimpljdbc.java
...
587. statement.setInt(1, memberID);
588. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 67:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
...
508. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
508. user = getOnlineUser(sessionID);
...
630. return user;
```



```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



```
284. public void prepareList(GenericRequest request) //watchwebhandler.java
...
287. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
291. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
284. public void prepareList(GenericRequest request) //watchwebhandler.java
     ...
     291. int memberID = onlineUser.getMemberID();
     ...
     294. Collection watchBeans = DAOFactory.getWatchDAO().getWatches_forMember
     (memberID);
```

```
650. public Collection getWatches_forMember(int memberID) //watchdaoimpljdbc.java
     ...
     665. statement.setInt(1, memberID);
     666. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 68:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
     843. String[] checkList = request.getParameterValues("selectedmember");
     ...
     851. int memberID = Integer.parseInt(checkList[i]);
     ...
     854. MemberBean memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     176. return memberDAO.getMember(memberID);
```

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
     ...
     1220. statement.setInt(1, memberID);
     1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 69:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
     ...
     843. String[] checkList = request.getParameterValues("selectedmember");
     ...
     869. int memberID = Integer.parseInt(checkList[i]);
```

```
...
872. DAOFactory.getMemberDAO().updateStatus(memberID,
MemberBean.MEMBER_STATUS_DISABLE);
```

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
...
2014. statement.setInt(2, memberID);
...
2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 70:

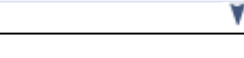
Query Name - **Parameter_Tampering**

Severity - **Serious**

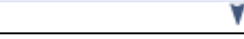
```
338. public void logout(GenericRequest request, GenericResponse
response) //onlineusermanager.java
...
344. String sessionID = request.getSessionId();
...
360. setOnlineUser(sessionID, user);
```

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
...
732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

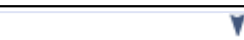
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
389. return getOnlineUser(request.getServletRequest());
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
615. String memberName = StringUtil.getEmptyStringIfNull
(authenticator.getRemoteUser(request));
```



```
62. public String getRemoteUser(GenericRequest
request) //samplerealmauthenticator.java
...
64. String memberName = request.getRemoteUser();
65. memberName = StringUtil.getEmptyStringIfNull(memberName);
```



```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
...
123. return str;
```

```
62. public String getRemoteUser(GenericRequest
```

```
request) //samplerealmauthenticator.java
    ...
    65. memberName = StringUtil.getEmptyStringIfNull(memberName);
    ...
    102. return memberName;
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    615. String memberName = StringUtil.getEmptyStringIfNull
    (authenticator.getRemoteUser(request));
```

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
    ...
    123. return str;
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    615. String memberName = StringUtil.getEmptyStringIfNull
    (authenticator.getRemoteUser(request));
    ...
    618. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    158. statement.setString(1, memberName);
    159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 71:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
338. public void logout(GenericRequest request, GenericResponse
response) //onlineusermanager.java
    ...
    344. String sessionID = request.getSessionId();
```

[Table of Contents] [Description]

Path 72:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
338. public void logout(GenericRequest request, GenericResponse
response) //onlineusermanager.java
    ...
```

```
344. String sessionID = request.getSessionId();
```

[Table of Contents] [Description]

Path 73:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
389. return getOnlineUser(request.getServletRequest());
```

▼

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
     ...
479. HttpSession session = request.getSession();
480. String sessionID = session.getId();
     ...
589. setOnlineUser(sessionID, user);
```

▼

```
730. private synchronized void setOnlineUser(String sessionID, OnlineUser
user) //onlineusermanager.java
     ...
732. OnlineUser onlineUser = getOnlineUser(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
430. String memberName = StringUtil.getEmptyStringIfNull(request.getRemoteUser
()));
```

▼

```
121. public static String getEmptyStringIfNull(String str) //stringutil.java
     ...
123. return str;
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
430. String memberName = StringUtil.getEmptyStringIfNull(request.getRemoteUser
());
     ...
433. DAOFactory.getMemberDAO().findByAlternateKey_MemberName(memberName);
```

▼

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
     ...
158. statement.setString(1, memberName);
159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 74:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      479. HttpSession session = request.getSession();
      480. String sessionID = session.getId();
      ...
      508. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      508. user = getOnlineUser(sessionID);
      ...
      630. return user;
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      389. return getOnlineUser(request.getServletRequest());
```

```
518. public void prepareEditEmail(GenericRequest request) //memberwebhandler.java
      ...
      521. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      525. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
518. public void prepareEditEmail(GenericRequest request) //memberwebhandler.java
      ...
      525. int memberID = onlineUser.getMemberID();
      526. MemberBean memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
        ...
        1220. statement.setInt(1, memberID);
        1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 75:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        392. String sessionID = request.getSessionId();
        ...
        420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
        ...
        727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        420. user = getOnlineUser(sessionID);
        ...
        465. return user;
```

▼

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
        ...
        65. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
        ...
        69. int memberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
        147. return memberID;
```

▼

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
        ...
        69. int memberID = onlineUser.getMemberID();
        ...
        72. int currentFavoriteCount = DAOFactory.getFavoriteThreadDAO
        ().getNumberOfFavoriteThreads_inMember(memberID);
```

▼

```
390. public int getNumberOfFavoriteThreads_inMember(int
memberID) //favoritethreaddaoimpljdbc.java
        ...
        403. statement.setInt(1, memberID);
```

```
404. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 76:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     392. String sessionID = request.getSessionId();
     ...
     420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     420. user = getOnlineUser(sessionID);
     ...
     465. return user;
```

▼

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
     ...
     95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     148. int memberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

▼

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
     ...
     148. int memberID = onlineUser.getMemberID();
     149. DAOFactory.getMemberForumDAO().create(memberID, forumID,
     MVNForumPermission.PERMISSION_FORUM_ADMIN);
```

▼

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
     ...
     124. DAOFactory.getMemberDAO().findByPrimaryKey(memberID);
```

▼

```
121. public void findByPrimaryKey(int memberID) throws ObjectNotFoundException,
```

```
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    122. memberDAO.findByPrimaryKey(memberID);
```

```
78. public void findByPrimaryKey(int memberID) //memberdaoimpljdbc.java
    ...
    91. statement.setInt(1, memberID);
    92. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 77:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
    250. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    254. String memberName = onlineUser.getMemberName();
```

```
150. public String getMemberName() //onlineuserimpl.java
    151. return memberName;
```

```
237. public void processAdd(GenericRequest request, GenericResponse
response) //postwebhandler.java
    ...
    254. String memberName = onlineUser.getMemberName();
    ...
    320. String lastPostMemberName = memberName;
    ...
    349. threadID = DAOFactory.getThreadDAO().createThread(forumID, memberName,
```

```
lastPostMemberName,
```

```
198. public int createThread(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
    206. create(forumID, memberName, lastPostMemberName, threadTopic, threadBody,
    threadVoteCount, threadVoteTotalStars, threadCreationDate, threadLastPostDate,
    threadType, threadPriority,threadOption, threadStatus, threadHasPoll,
    threadViewCount, threadReplyCount, threadIcon, threadDuration,
    threadAttachCount);
```

```
109. private void create(int forumID, String memberName, String
lastPostMemberName, //threaddaoimpljdbc.java
    ...
    144. DAOFactory.getMemberDAO().findByAlternateKey_MemberName
    (lastPostMemberName);
```

```
113. public String findByAlternateKey_MemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    114. return memberDAO.findByAlternateKey_MemberName(memberName);
```

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    158. statement.setString(1, memberName);
    159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 78:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
```

[Table of Contents] [Description]

Path 79:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
      ...
      583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      587. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
      ...
      587. int memberID = onlineUser.getMemberID();
      ...
      624. DAOFactory.getMemberDAO().updatePassword(memberID, // primary key
```

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      563. statement.setInt(2, memberID);
      ...
      573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 80:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
```

```
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
    ...
    536. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    541. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
    147. return memberID;
```

```
530. public void processUpdateEmail(GenericRequest request) //memberwebhandler.java
    ...
    541. int memberID = onlineUser.getMemberID();
    ...
    563. DAOFactory.getMemberDAO().updateActivateCode(memberID, "");
```

```
295. public void updateActivateCode(int memberID, String memberActivateCode) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    296. memberDAO.updateActivateCode(memberID, memberActivateCode);
```

```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
    ...
    667. statement.setInt(2, memberID);
    ...
    669. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 81:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
      ...
      583. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      587. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
577. public void processUpdatePassword(GenericRequest
request) //memberwebhandler.java
      ...
      587. int memberID = onlineUser.getMemberID();
      ...
      613. String currentPassword = DAOFactory.getMemberDAO().getPassword(memberID);
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      256. return memberDAO.getPassword(memberID);
```

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
      ...
      904. statement.setInt(1, memberID);
      905. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 82:

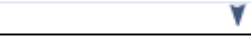
Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
        ...
        727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        420. user = getOnlineUser(sessionID);
        ...
        465. return user;
```

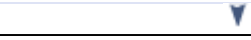


```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
        ...
        95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
        ...
        150. onlineUser.reloadPermission();
```

```
179. public void reloadPermission() //onlineuserimpl.java
        ...
        185. MemberBean memberBean = MemberCache.getInstance().getMember(memberID);
```

```
109. public MemberBean getMember(int memberID) //membercache.java
        ...
        118. memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
        176. return memberDAO.getMember(memberID);
```



```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
        ...
        1220. statement.setInt(1, memberID);
        1221. resultSet = statement.executeQuery();
```

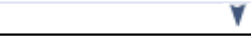
[Table of Contents] [Description]

Path 83:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
        ...
        392. String sessionID = request.getSessionId();
        ...
        420. user = getOnlineUser(sessionID);
```



```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

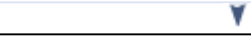
```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

```
351. public static void ensureCorrectCurrentPassword(GenericRequest request, String
password, String passwordMD5) //myutil.java
    ...
    361. OnlineUser onlineUser = OnlineUserManager.getInstance().getOnlineUser
    (request);
    ...
    363. String memberName = onlineUser.getMemberName();
```

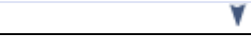
```
150. public String getMemberName() //onlineuserimpl.java
    151. return memberName;
```

```
351. public static void ensureCorrectCurrentPassword(GenericRequest request, String
password, String passwordMD5) //myutil.java
    ...
    363. String memberName = onlineUser.getMemberName();
    ...
    367. onlineUserFactory.ensureCorrectPassword(memberName,
    OnlineUserManager.PASSWORD_OF_METHOD_REALM, true);
```

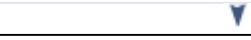
```
353. public void ensureCorrectPassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
    ...
    356. boolean isCorrectPassword = validatePassword(loginName, password,
    isEncodedPassword);
```



```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
    ...
    322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (loginName);
```



```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```



```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
```

```
...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 84:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
392. String sessionID = request.getSessionId();
...
420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
...
727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
...
420. user = getOnlineUser(sessionID);
...
465. return user;
```

▼

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
...
95. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
...
148. int memberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
147. return memberID;
```

▼

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
...
148. int memberID = onlineUser.getMemberID();
149. DAOFactory.getMemberForumDAO().create(memberID, forumID,
MVNForumPermission.PERMISSION_FORUM_ADMIN);
```

▼

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
...
147. statement.setInt(1, memberID);
```

```
148. statement.setInt(2, forumID);
149. statement.setInt(3, permission);
...
151. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 85:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    392. String sessionID = request.getSessionId();
    ...
    420. user = getOnlineUser(sessionID);
```

▼

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
    ...
    727. return (OnlineUser)userMap.get(sessionID);
```

▼

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
    ...
    420. user = getOnlineUser(sessionID);
    ...
    465. return user;
```

▼

```
118. public void prepareList(GenericRequest request) //messagefolderwebhandler.java
    ...
    129. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
    ...
    134. int memberID = onlineUser.getMemberID();
```

▼

```
146. public int getMemberID() //onlineuserimpl.java
    147. return memberID;
```

▼

```
118. public void prepareList(GenericRequest request) //messagefolderwebhandler.java
    ...
    134. int memberID = onlineUser.getMemberID();
    ...
    136. Collection messageFolderBeans = DAOFactory.getMessageFolderDAO
    ().getMessageFolders_inMember(memberID);
```

▼

```
244. public Collection getMessageFolders_inMember(int
memberID) //messagefolderdaoimpljdbc.java
    ...
    259. statement.setInt(1, memberID);
    260. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 86:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     392. String sessionID = request.getSessionId();
     ...
     420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
     ...
     727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
     ...
     420. user = getOnlineUser(sessionID);
     ...
     465. return user;
```

```
84. public void processAdd(GenericRequest request) //messagefolderwebhandler.java
     ...
     98. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
     ...
     110. int folderOrder = DAOFactory.getMessageFolderDAO().getMaxFolderOrder
     (onlineUser.getMemberID());
```

```
146. public int getMemberID() //onlineuserimpl.java
     147. return memberID;
```

```
84. public void processAdd(GenericRequest request) //messagefolderwebhandler.java
     ...
     110. int folderOrder = DAOFactory.getMessageFolderDAO().getMaxFolderOrder
     (onlineUser.getMemberID());
```

```
284. public int getMaxFolderOrder(int memberID) //messagefolderdaoimpljdbc.java
     ...
     299. statement.setInt(1, memberID);
     300. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 87:

Query Name - **Parameter_Tampering**

Severity

- Serious

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      392. String sessionID = request.getSessionId();
      ...
      420. user = getOnlineUser(sessionID);
```

```
725. private synchronized OnlineUser getOnlineUser(String
sessionID) //onlineusermanager.java
      ...
      727. return (OnlineUser)userMap.get(sessionID);
```

```
385. public OnlineUser getOnlineUser(GenericRequest
request) //onlineusermanager.java
      ...
      420. user = getOnlineUser(sessionID);
      ...
      465. return user;
```

```
284. public void prepareList(GenericRequest request) //watchwebhandler.java
      ...
      287. OnlineUser onlineUser = onlineUserManager.getOnlineUser(request);
      ...
      291. int memberID = onlineUser.getMemberID();
```

```
146. public int getMemberID() //onlineuserimpl.java
      147. return memberID;
```

```
284. public void prepareList(GenericRequest request) //watchwebhandler.java
      ...
      291. int memberID = onlineUser.getMemberID();
      ...
      294. Collection watchBeans = DAOFactory.getWatchDAO().getWatches_forMember
      (memberID);
```

```
650. public Collection getWatches_forMember(int memberID) //watchdaoimpljdbc.java
      ...
      665. statement.setInt(1, memberID);
      666. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 88:

Query Name - **Parameter_Tampering**

Severity - Serious

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
```

```
...
532. Cookie[] cookies = request.getCookies();
...
535. Cookie cookie = cookies[i];
...
538. memberName = Encoder.decodeURL(cookie.getValue());
```



```
160. public static String decodeURL(String input) //encoder.java
...
162. return URLDecoder.decode(input, "UTF-8");
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
538. memberName = Encoder.decodeURL(cookie.getValue());
...
548. user = login(request, null, memberName, encodedPassword, true);
```



```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
(request, response, memberName, encodedPassword, true);
```



```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
...
70. String loginName, String password,
...
76. return getAuthenticatedUser(req, null, loginName, password,
isEncodedPassword);
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
...
81. String loginName, String password,
...
123. if (validatePassword(loginName, password, isEncodedPassword) == false) {
```



```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
...
322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(loginName);
```



```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      1416. statement.setString(1, memberName);
      1417. resultSet = statement.executeQuery();
      ...
      1421. return resultSet.getInt(1);
```



```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      191. return memberDAO.getMemberIDFromMemberName(memberName);
```



```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
      ...
      322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
      (loginName);
      ...
      344. return encodedPassword.equals(DAOFactory.getMemberDAO().getPassword
      (memberId));
```



```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      256. return memberDAO.getPassword(memberID);
```



```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
      ...
      904. statement.setInt(1, memberID);
      905. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 89:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      532. Cookie[] cookies = request.getCookies();
      ...
      535. Cookie cookie = cookies[i];
      ...
      538. memberName = Encoder.decodeURL(cookie.getValue());
```

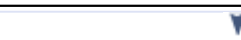


```
160. public static String decodeURL(String input) //encoder.java
      ...
      162. return URLDecoder.decode(input, "UTF-8");
```

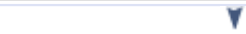


```
468. public OnlineUser getOnlineUser(HttpServletRequest
```

```
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
    ...
    548. user = login(request, null, memberName, encodedPassword, true);
```

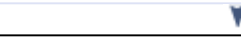


```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```



```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    81. String loginName, String password,
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
    141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (memberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    1416. statement.setString(1, memberName);
    1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 90:

Query Name - **Parameter_Tampering**

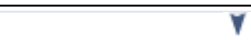
Severity - **Serious**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
```

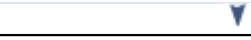
```
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
```

```
160. public static String decodeURL(String input) //encoder.java
    ...
    162. return URLDecoder.decode(input, "UTF-8");
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
    ...
    548. user = login(request, null, memberName, encodedPassword, true);
```

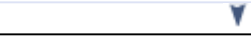


```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```

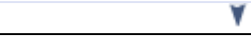


```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```

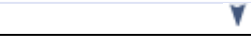
```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    81. String loginName, String password,
    ...
    123. if (validatePassword(loginName, password, isEncodedPassword) == false) {
```



```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
    ...
    322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (loginName);
```



```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```



```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
```

```
...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
...
1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
...
322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(loginName);
...
344. return encodedPassword.equals(DAOFactory.getMemberDAO().getPassword
(memberId));
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
256. return memberDAO.getPassword(memberID);
```

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
...
904. statement.setInt(1, memberID);
905. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 91:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
538. memberName = Encoder.decodeURL(cookie.getValue());
```

```
160. public static String decodeURL(String input) //encoder.java
...
162. return URLDecoder.decode(input, "UTF-8");
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
...
538. memberName = Encoder.decodeURL(cookie.getValue());
...
548. user = login(request, null, memberName, encodedPassword, true);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
      194. String memberName, String memberPassword, boolean passwordEncoded)
      ...
      211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
      (request, response, memberName, encodedPassword, true);
```



```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
      ...
      70. String loginName, String password,
      ...
      76. return getAuthenticatedUser(req, null, loginName, password,
      isEncodedPassword);
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
      ...
      114. if (DAOFactory.getMemberDAO().getActivateCode(memberID).equals
      (MemberBean.MEMBER_ACTIVATECODE_ACTIVATED) == false) {
```



```
137. public String getActivateCode(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      138. return memberDAO.getActivateCode(memberID);
```



```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
      1000. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 92:

Query Name - **Parameter_Tampering**

Severity - Serious

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      538. memberName = Encoder.decodeURL(cookie.getValue());
```



```
160. public static String decodeURL(String input) //encoder.java
      ...
      162. return URLDecoder.decode(input, "UTF-8");
```



```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
      ...
      538. memberName = Encoder.decodeURL(cookie.getValue());
      ...
      548. user = login(request, null, memberName, encodedPassword, true);
```



```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
      194. String memberName, String memberPassword, boolean passwordEncoded)
      ...
      211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
      (request, response, memberName, encodedPassword, true);
```



```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
      ...
      70. String loginName, String password,
      ...
      76. return getAuthenticatedUser(req, null, loginName, password,
      isEncodedPassword);
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
```

```
...
94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
...
104. MemberBean memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    176. return memberDAO.getMember(memberID);
```

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
    ...
    1220. statement.setInt(1, memberID);
    1221. resultSet = statement.executeQuery();
```

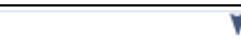
[Table of Contents] [Description]

Path 93:

Query Name - **Parameter_Tampering**

Severity - **Serious**

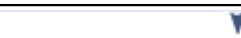
```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
```



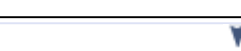
```
160. public static String decodeURL(String input) //encoder.java
    ...
    162. return URLDecoder.decode(input, "UTF-8");
```

```
468. public OnlineUser getOnlineUser(HttpServletRequest
request) //onlineusermanager.java
    ...
    538. memberName = Encoder.decodeURL(cookie.getValue());
    ...
    548. user = login(request, null, memberName, encodedPassword, true);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```



```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
      (memberName);
```



```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      191. return memberDAO.getMemberIDFromMemberName(memberName);
```



```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      1416. statement.setString(1, memberName);
      1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 94:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      694. String[] unreadList = request.getParameterValues("selectedmessage");
      ...
      696. int messageID = Integer.parseInt(unreadList[i]);
      ...
      699. messageBean = DAOFactory.getMessageDAO().getMessage(messageID);
```



```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java
      ...
      586. statement.setInt(1, messageID);
      587. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 95:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
620. public void processMessage(GenericRequest request) //messagewebhandler.java
      ...
      718. String[] movingList = request.getParameterValues("selectedmessage");
```

```
...
723. int messageID = Integer.parseInt(movingList[i]);
...
726. messageBean = DAOFactory.getMessageDAO().getMessage(messageID);
```



```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java

...
586. statement.setInt(1, messageID);
587. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 96:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java

...
59. String ret = request.getParameter(param);
...
61. return ret.trim();
```



```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java

...
891. String memberName = GenericParamUtil.getParameter(request, "MemberName");
...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java

...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```



```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java

...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
906. MemberBean bean = DAOFactory.getMemberDAO().getMember(memberID);
```



```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
176. return memberDAO.getMember(memberID);
```



```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java

...
```

```
1220. statement.setInt(1, memberID);
1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 97:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
59. String ret = request.getParameter(param);
    ...
61. return ret.trim();
```

▼

```
880. public void forgotPassword(GenericRequest request) //memberwebhandler.java
    ...
891. String memberName = GenericParamUtil.getParameter(request, "MemberName");
    ...
901. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

▼

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
```

▼

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

▼

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 98:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
59. String ret = request.getParameter(param);
    ...
61. return ret.trim();
```

▼

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    642. String order = GenericParamUtil.getParameter(request, "order");
    ...
    671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
    (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1468. return getBeans_withSortSupport_limit_mysql(offset, rowsToReturn, sort,
    order, memberStatus);
```

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1541. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
    1542. sql.append(" LIMIT ?, ?");
    ...
    1546. statement = connection.prepareStatement(sql.toString());
    1547. statement.setInt(1, offset);
    1548. statement.setInt(2, rowsToReturn);
    1549. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 99:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
    59. String ret = request.getParameter(param);
    ...
    61. return ret.trim();
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    642. String order = GenericParamUtil.getParameter(request, "order");
    ...
    671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
    (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1470. return getBeans_withSortSupport_limit_noscroll(offset, rowsToReturn,
    sort, order, memberStatus);
```



```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
```

```
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1672. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
    ...
1676. statement = connection.prepareStatement(sql.toString());
1677. statement.setMaxRows(offset + rowsToReturn);
1678. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 100:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
59. String ret = request.getParameter(param);
    ...
61. return ret.trim();
```

▼

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
642. String order = GenericParamUtil.getParameter(request, "order");
    ...
671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
(offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

▼

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1472. return getBeans_withSortSupport_limit_general(offset, rowsToReturn,
sort, order, memberStatus);
```

▼

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1807. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
    ...
1811. statement = connection.prepareStatement(sql.toString(),
ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
1812. statement.setMaxRows(offset + rowsToReturn);
    ...
1814. statement.setFetchSize(Math.min(rowsToReturn, DBUtils.MAX_FETCH_SIZE));
    ...
1818. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 101:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```



```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
      ...
      116. int groupID = GenericParamUtil.getParameterInt(request, "group");
      ...
      124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```



```
89. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
      ...
      111. statement.setInt(1, groupID);
      112. statement.setInt(2, permission);
      ...
      114. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 102:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
```

```
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
...
119. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```



```
88. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
121. statement.setInt(1, memberID);
122. statement.setInt(2, permission);
...
124. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 103:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
111. public void processUpdate(GenericRequest request) //groupforumwebhandler.java
...
134. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
145. DAOFactory.getGroupForumDAO().create(groupID, forumID, perm);
```



```
105. public void create(int groupID, int forumID, int
permission) //groupforumdaoimpljdbc.java
...
145. statement.setInt(1, groupID);
```

```
146. statement.setInt(2, forumID);
147. statement.setInt(3, permission);
...
149. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 104:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

▼

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

▼

```
92. public void processUpdate(GenericRequest
request) //grouppermissionwebhandler.java
...
116. int groupID = GenericParamUtil.getParameterInt(request, "group");
...
124. DAOFactory.getGroupPermissionDAO().create(groupID, perm);
```

▼

```
104. public void create(int groupID, int
permission) //grouppermissiondaoimpljdbc.java
...
136. statement.setInt(1, groupID);
137. statement.setInt(2, permission);
...
139. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 105:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
```

```
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
...
243. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
254. DAOFactory.getMemberForumDAO().create(memberID, forumID, perm);
```

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
...
124. DAOFactory.getMemberDAO().findByPrimaryKey(memberID);
```

```
78. public void findByPrimaryKey(int memberID) //memberdaoimpljdbc.java
...
91. statement.setInt(1, memberID);
92. resultSet = statement.executeQuery();
```

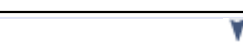
[Table of Contents] [Description]

Path 106:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
...
91. String ret = getParameter(request, param, checkEmpty);
...
98. return ret;
```

```
89. public void processAdd(GenericRequest request, GenericResponse
response) //forumwebhandler.java
      ...
      103. String forumOwnerName = GenericParamUtil.getParameterSafe(request,
      "ForumOwnerName", false);
      ...
      131. forumOwnerName = DAOFactory.getMemberDAO().findByAlternateKey_MemberName
      (forumOwnerName);
```

▼

```
113. public String findByAlternateKey_MemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      114. return memberDAO.findByAlternateKey_MemberName(memberName);
```

▼

```
141. public String findByAlternateKey_MemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      158. statement.setString(1, memberName);
      159. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 107:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

▼

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
      ...
      91. String ret = getParameter(request, param, checkEmpty);
      ...
      98. return ret;
```

▼

```
273. public static String getParameterEmail(GenericRequest request, String
param) //genericparamutil.java
      ...
      276. String email = getParameterSafe(request, param, true);
      ...
      278. return email;
```

▼

```
103. public void processAdd(GenericRequest request) //memberwebhandler.java
      ...
      153. String memberEmail = GenericParamUtil.getParameterEmail(request,
```

```
"MemberEmail");
...
184. memberEmail, memberEmailVisible, memberNameVisible,
```

```
63. public void create(String memberName, String memberPassword, String
memberFirstEmail, String memberEmail, int memberEmailVisible, int memberNameVisible,
String memberFirstIP, String memberLastIP, int memberViewCount, int memberPostCount,
Timestamp memberCreationDate, Timestamp memberModifiedDate, Timestamp
memberExpireDate, Timestamp memberPasswordExpireDate, Timestamp memberLastLogon, int
memberOption, int memberStatus, String memberActivateCode, String
memberTempPassword, int memberMessageCount, int memberMessageOption, int
memberPostsPerPage, int memberWarnCount, int memberVoteCount, int
memberVoteTotalStars, int memberRewardPoints, String memberTitle, double
memberTimeZone, String memberSignature, String memberAvatar, String memberSkin,
String memberLanguage, String memberFirstname, String memberLastname, int
memberGender, Date memberBirthday, String memberAddress, String memberCity, String
memberState, String memberCountry, String memberPhone, String memberMobile, String
memberFax, String memberCareer, String memberHomepage, String memberYahoo, String
memberAol, String memberIcq, String memberMsn, String memberCoolLink1, String
memberCoolLink2) //memberdaoexternaluserdatabaseabstract.java
```

```
...
66. memberDAO.create(memberName, memberPassword, memberFirstEmail,
memberEmail, memberEmailVisible, memberNameVisible, memberFirstIP,
memberLastIP, memberViewCount, memberPostCount, memberCreationDate,
memberModifiedDate, memberExpireDate, memberPasswordExpireDate,
memberLastLogon, memberOption, memberStatus, memberActivateCode,
memberTempPassword, memberMessageCount, memberMessageOption,
memberPostsPerPage, memberWarnCount, memberVoteCount, memberVoteTotalStars,
memberRewardPoints, memberTitle, memberTimeZone, memberSignature,
memberAvatar, memberSkin, memberLanguage, memberFirstname, memberLastname,
memberGender, memberBirthday, memberAddress, memberCity, memberState,
memberCountry, memberPhone, memberMobile, memberFax, memberCareer,
memberHomepage, memberYahoo, memberAol, memberIcq, memberMsn, memberCoolLink1,
memberCoolLink2);
```

```
259. public void create(String memberName, String memberPassword, String
memberFirstEmail, //memberdaoimpljdbc.java
```

```
260. String memberEmail, int memberEmailVisible, int memberNameVisible,
...
293. findByAlternateKey_MemberEmail(memberEmail);
```

```
177. public void findByAlternateKey_MemberEmail(String
memberEmail) //memberdaoimpljdbc.java
```

```
...
194. statement.setString(1, memberEmail);
195. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 108:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
```

```
...
73. ret = ret.trim();
...
79. return ret;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
...
1010. String memberName = GenericParamUtil.getParameter(request, "member",
true);
...
1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
...
1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1047. DAOFactory.getMemberDAO().updatePassword(memberID, memberPassword,
passwordExpireDate);
```

```
540. public void updatePassword(int memberID, // primary
key //memberdaoimpljdbc.java
...
563. statement.setInt(2, memberID);
...
573. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 109:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
...
```

```
1010. String memberName = GenericParamUtil.getParameter(request, "member",
true);
...
1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java

...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java

...
1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1048. DAOFactory.getMemberDAO().updateTempPassword(memberID, "");// reset the
temp password
```

```
602. public void updateTempPassword(int memberID, // primary
key //memberdaoimpljdbc.java

...
619. statement.setInt(2, memberID);
...
621. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

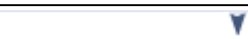
Path 110:

Query Name - **Parameter_Tampering**

Severity - **Serious**

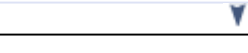
```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java

...
1059. String memberName = GenericParamUtil.getParameter(request, "MemberName",
true);
...
1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
```

```
...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```



```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
...
1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1081. SendMailUtil.sendActivationCodeEmail(memberID, serverName);
```



```
73. public static void sendActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
...
76. MailMessageStruct mailMessageStruct = getActivationCodeEmail(memberID,
serverName);
```



```
85. public static MailMessageStruct getActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
...
106. DAOFactory.getMemberDAO().updateActivateCode(memberID, activateCode);
```



```
295. public void updateActivateCode(int memberID, String memberActivateCode) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
296. memberDAO.updateActivateCode(memberID, memberActivateCode);
```



```
650. public void updateActivateCode(int memberID, // primary
key //memberdaoimpljdbc.java
...
667. statement.setInt(2, memberID);
...
669. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 111:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
      ...
      1010. String memberName = GenericParamUtil.getParameter(request, "member",
      true);
      ...
      1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
```

▼

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```

▼

```
1003. public void resetPassword(GenericRequest request) //memberwebhandler.java
      ...
      1018. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
      ...
      1024. String currentTempPassword = DAOFactory.getMemberDAO().getTempPassword
      (memberID);
```

▼

```
939. public String getTempPassword(int memberID) //memberdaoimpljdbc.java
      ...
      999. statement.setInt(1, memberID);
```

[Table of Contents] [Description]

Path 112:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

▼

```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
      ...
      1059. String memberName = GenericParamUtil.getParameter(request, "MemberName",
      true);
      ...
      1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
      (memberName);
```

▼

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
```

```
...
137. String key = new String("getMemberIDFromMemberName_" + memberName);
138. Integer memberIDInterger = (Integer)cache.retrieve(key);
...
144. memberID = memberIDInterger.intValue();
...
154. return memberID;
```



```
1051. public void sendActivateCode(GenericRequest request) //memberwebhandler.java
...
1065. memberID = MemberCache.getInstance().getMemberIDFromMemberName
(memberName);
...
1081. SendMailUtil.sendActivationCodeEmail(memberID, serverName);
```



```
73. public static void sendActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
...
76. MailMessageStruct mailMessageStruct = getActivationCodeEmail(memberID,
serverName);
```



```
85. public static MailMessageStruct getActivationCodeEmail(int memberID, String
serverName) //sendmailutil.java
...
90. String activateCode = DAOFactory.getMemberDAO().getActivateCode(memberID);
```



```
137. public String getActivateCode(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
138. return memberDAO.getActivateCode(memberID);
```



```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
...
999. statement.setInt(1, memberID);
1000. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 113:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```



```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
```

```
...
376. String strMemberName = GenericParamUtil.getParameter(request, "member",
false);
...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(strMemberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    1416. statement.setString(1, memberName);
    1417. resultSet = statement.executeQuery();
    ...
    1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
    ...
    389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
    (strMemberName);
    ...
    398. memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```

```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
    ...
    1220. statement.setInt(1, memberID);
    1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 114:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
```

```
...
376. String strMemberName = GenericParamUtil.getParameter(request, "member",
false);
...
389. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(strMemberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
    ...
    1416. statement.setString(1, memberName);
    1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 115:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
88. public static String getParameterSafe(GenericRequest request, String param,
boolean checkEmpty) //genericparamutil.java
    ...
    91. String ret = getParameter(request, param, checkEmpty);
    ...
    98. return ret;
```

```
273. public static String getParameterEmail(GenericRequest request, String
param) //genericparamutil.java
    ...
    276. String email = getParameterSafe(request, param, true);
    ...
    278. return email;
```

```
366. public void prepareView(GenericRequest request) //memberwebhandler.java
    ...
    403. String memberEmail = GenericParamUtil.getParameterEmail(request,
    "memberemail");// check for better security
    ...
    405. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberEmail
```

```
(memberEmail);
```

```
1436. public final int getMemberIDFromMemberEmail(String
memberEmail) //memberdaoimpljdbc.java
...
1446. statement.setString(1, memberEmail);
1447. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 116:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java
...
119. int retValue = getParameterInt(request, param);
...
125. return retValue;
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
...
650. offset = GenericParamUtil.getParameterUnsignedInt(request, "offset");
...
671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
(offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
...
1468. return getBeans_withSortSupport_limit_mysql(offset, rowsToReturn, sort,
order, memberStatus);
```

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1547. statement.setInt(1, offset);
      1548. statement.setInt(2, rowsToReturn);
      1549. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 117:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      119. int retValue = getParameterInt(request, param);
      ...
      125. return retValue;
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
      ...
      650. offset = GenericParamUtil.getParameterUnsignedInt(request, "offset");
      ...
      671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
      (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1470. return getBeans_withSortSupport_limit_noscroll(offset, rowsToReturn,
      sort, order, memberStatus);
```

```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1677. statement.setMaxRows(offset + rowsToReturn);
      1678. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 118:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
      ...
      71. String ret = request.getParameter(param);
      ...
      73. ret = ret.trim();
      ...
      79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      104. String inputStr = getParameter(request, param, true);
      ...
      107. ret = Integer.parseInt(inputStr);
      ...
      113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java
      ...
      119. int retValue = getParameterInt(request, param);
      ...
      125. return retValue;
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
      ...
      650. offset = GenericParamUtil.getParameterUnsignedInt(request, "offset");
      ...
      671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
      (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1472. return getBeans_withSortSupport_limit_general(offset, rowsToReturn,
      sort, order, memberStatus);
```

```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
```

```
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
1812. statement.setMaxRows(offset + rowsToReturn);
    ...
1814. statement.setFetchSize(Math.min(rowsToReturn, DBUtils.MAX_FETCH_SIZE));
    ...
1818. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 119:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
71. String ret = request.getParameter(param);
    ...
73. ret = ret.trim();
    ...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
104. String inputStr = getParameter(request, param, true);
    ...
107. ret = Integer.parseInt(inputStr);
    ...
113. return ret;
```

```
240. public void processUpdateMemberStatus(GenericRequest
request) //memberwebhandler.java
    ...
248. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
    ...
267. DAOFactory.getMemberDAO().updateStatus(memberID, memberStatus);
```

```
1997. public void updateStatus(int memberID, // primary key //memberdaoimpljdbc.java
    ...
2014. statement.setInt(2, memberID);
    ...
2016. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 120:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
```

```
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java

...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java

...
119. int retValue = getParameterInt(request, param);
...
125. return retValue;
```

```
702. public void prepareListUserExpire(GenericRequest
request) //memberwebhandler.java

...
719. offset = GenericParamUtil.getParameterUnsignedInt(request, "offset");
...
731. Collection expireMemberBeans = DAOFactory.getMemberDAO
().getMembers_inExpire_limit(expiresoonDate, offset, postsPerPage, sort,
order);
```

```
2129. public Collection getMembers_inExpire_limit(Timestamp expireDate, int offset,
int rowsToReturn, String sort, String order) //memberdaoimpljdbc.java

...
2172. statement.setMaxRows(offset + rowsToReturn);
2173. statement.setTimestamp(1, expireDate);
...
2175. statement.setFetchSize(Math.min(rowsToReturn, DBUtils.MAX_FETCH_SIZE));
...
2179. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 121:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java

...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
```

```
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
116. public static int getParameterUnsignedInt(GenericRequest request, String
param) //genericparamutil.java
...
119. int retValue = getParameterInt(request, param);
...
125. return retValue;
```

```
829. public void updateMemberExpireProcess(GenericRequest
request) //memberwebhandler.java
...
841. int expiredDays = GenericParamUtil.getParameterUnsignedInt(request,
"addtime");
...
848. long offsetDate = expiredDays * DateUtil.DAY;
...
859. Timestamp memberExpireDate = new Timestamp(expireDate.getTime() +
offsetDate);
860. DAOFactory.getMemberDAO().updateMember_expireDate(memberID,
memberExpireDate);
```

```
2237. public void updateMember_expireDate(int memberID, // primary
key //memberdaoimpljdbc.java
2238. Timestamp expireDate)
...
2251. statement.setTimestamp(1, expireDate);
...
2254. statement.setInt(2, memberID);
...
2256. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 122:

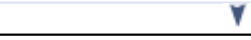
Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
```

```
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```



```
220. public void processUpdate(GenericRequest request) //memberforumwebhandler.java
...
243. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
254. DAOFactory.getMemberForumDAO().create(memberID, forumID, perm);
```

```
105. public void create(int memberID, int forumID, int
permission) //memberforumdaoimpljdbc.java
...
147. statement.setInt(1, memberID);
148. statement.setInt(2, forumID);
149. statement.setInt(3, permission);
...
151. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 123:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
95. public void processUpdate(GenericRequest
request) //memberpermissionwebhandler.java
...
```

```
119. int memberID = GenericParamUtil.getParameterInt(request, "memberid");
...
127. DAOFactory.getMemberPermissionDAO().create(memberID, perm);
```

```
104. public void create(int memberID, int
permission) //memberpermissiondaoimpljdbc.java
...
138. statement.setInt(1, memberID);
139. statement.setInt(2, permission);
...
141. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 124:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
...
71. String ret = request.getParameter(param);
...
73. ret = ret.trim();
...
79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
70. public void prepareAdd(GenericRequest request, GenericResponse
response) //messagewebhandler.java
...
94. parentMessageID = GenericParamUtil.getParameterInt(request, "parent");
...
103. parentMessageBean = DAOFactory.getMessageDAO().getMessage
(parentMessageID);
```

```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java
...
586. statement.setInt(1, messageID);
587. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 125:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
    ...
    104. String inputStr = getParameter(request, param, true);
    ...
    107. ret = Integer.parseInt(inputStr);
    ...
    113. return ret;
```

```
746. public void processSendMessage(GenericRequest request) //messagewebhandler.java
    ...
    764. int messageID = GenericParamUtil.getParameterInt(request, "message");
    ...
    843. PrivateMessageUtil.deleteMessageInDatabase(messageID, logonMemberID);
```

```
68. public static void deleteMessageInDatabase(int messageID, int
memberID) //privatemessageutil.java
    ...
    77. DAOFactory.getPmAttachMessageDAO().delete_inMessage(messageID);
```

```
193. public void delete_inMessage(int messageID) //pmattachmessagedaoimpljdbc.java
    ...
    205. statement.setInt(1, messageID);
    206. statement.executeUpdate();
```

[Table of Contents] [Description]

Path 126:

Query Name - **Parameter_Tampering**

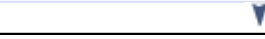
Severity - **Serious**

```
68. public static String getParameter(GenericRequest request, String param, boolean
checkEmpty) //genericparamutil.java
    ...
    71. String ret = request.getParameter(param);
    ...
    73. ret = ret.trim();
    ...
    79. return ret;
```

```
101. public static int getParameterInt(GenericRequest request, String
param) //genericparamutil.java
```

```
...
104. String inputStr = getParameter(request, param, true);
...
107. ret = Integer.parseInt(inputStr);
...
113. return ret;
```

```
61. public void processAdd(GenericRequest request) //favoritethreadwebhandler.java
...
83. int threadID = GenericParamUtil.getParameterInt(request, "thread");
...
92. threadBean = DAOFactory.getThreadDAO().getThread(threadID);
```



```
480. public ThreadBean getThread(int threadID) //threaddaoimpljdbc.java
...
493. statement.setInt(1, threadID);
494. resultSet = statement.executeQuery();
```

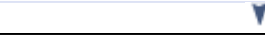
[Table of Contents] [Description]

Path 127:

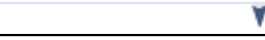
Query Name - **Parameter_Tampering**

Severity - **Serious**

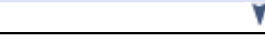
```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```



```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
...
130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
...
140. processLogin(request, response, memberName, memberPassword,
memberPasswordMD5);
```



```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
147. String memberName, String memberPassword, String memberPasswordMD5)
...
166. user = login(request, response, memberName, memberPassword, false);
```



```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
(request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
      ...
      70. String loginName, String password,
      ...
      76. return getAuthenticatedUser(req, null, loginName, password,
      isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      81. String loginName, String password,
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
      ...
      137. String key = new String("getMemberIDFromMemberName_" + memberName);
      138. Integer memberIDInterger = (Integer)cache.retrieve(key);
      ...
      144. memberID = memberIDInterger.intValue();
      ...
      154. return memberID;
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
      ...
      94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
      ...
      132. DAOFactory.getMemberDAO().updateLastLogon(memberID, now,
      request.getRemoteAddr());
```

```
317. public void updateLastLogon(int memberID, Timestamp memberLastLogon, String
lastIP) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      318. memberDAO.updateLastLogon(memberID, memberLastLogon, lastIP);
```

```
842. public void updateLastLogon(int memberID, // primary
key //memberdaoimpljdbc.java
      ...
      860. statement.setInt(3, memberID);
      ...
      862. if (statement.executeUpdate() != 1) {
```

[Table of Contents] [Description]

Path 128:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
```

```
...
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
...
130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
...
140. processLogin(request, response, memberName, memberPassword,
memberPasswordMD5);
```

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
147. String memberName, String memberPassword, String memberPasswordMD5)
...
166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
(request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
...
70. String loginName, String password,
...
76. return getAuthenticatedUser(req, null, loginName, password,
isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
...
81. String loginName, String password,
...
123. if (validatePassword(loginName, password, isEncodedPassword) == false) {
```

```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
...
322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(loginName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
      ...
      1416. statement.setString(1, memberName);
      1417. resultSet = statement.executeQuery();
      ...
      1421. return resultSet.getInt(1);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
318. public boolean validatePassword(String loginName, String password, boolean
isEncodedPassword) //onlineuserfactoryimpl.java
      ...
      322. int memberId = DAOFactory.getMemberDAO().getMemberIDFromMemberName
      (loginName);
      ...
      344. return encodedPassword.equals(DAOFactory.getMemberDAO().getPassword
      (memberId));
```

```
255. public String getPassword(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
      256. return memberDAO.getPassword(memberID);
```

```
891. public String getPassword(int memberID) //memberdaoimpljdbc.java
      ...
      904. statement.setInt(1, memberID);
      905. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 129:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
      ...
      114. String ret = request.getParameter(param);
      ...
      116. ret = ret.trim();
      ...
      122. return ret;
```

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
      ...
      130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
      ...
      140. processLogin(request, response, memberName, memberPassword,
```

```
memberPasswordMD5);
```

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    147. String memberName, String memberPassword, String memberPasswordMD5)
    ...
    166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    81. String loginName, String password,
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
    ...
    137. String key = new String("getMemberIDFromMemberName_" + memberName);
    138. Integer memberIDInterger = (Integer)cache.retrieve(key);
    ...
    144. memberID = memberIDInterger.intValue();
    ...
    154. return memberID;
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
    ...
    114. if (DAOFactory.getMemberDAO().getActivateCode(memberID).equals
    (MemberBean.MEMBER_ACTIVATECODE_ACTIVATED) == false) {
```

```
137. public String getActivateCode(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
    138. return memberDAO.getActivateCode(memberID);
```

```
986. public String getActivateCode(int memberID) //memberdaoimpljdbc.java
```

```
...
999. statement.setInt(1, memberID);
1000. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 130:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
    ...
    114. String ret = request.getParameter(param);
    ...
    116. ret = ret.trim();
    ...
    122. return ret;
```

```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
    ...
    130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
    ...
    140. processLogin(request, response, memberName, memberPassword,
    memberPasswordMD5);
```

```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    147. String memberName, String memberPassword, String memberPasswordMD5)
    ...
    166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
    194. String memberName, String memberPassword, boolean passwordEncoded)
    ...
    211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
    (request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
    ...
    70. String loginName, String password,
    ...
    76. return getAuthenticatedUser(req, null, loginName, password,
    isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
    ...
    81. String loginName, String password,
    ...
```

```
94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```



```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
     ...
     137. String key = new String("getMemberIDFromMemberName_" + memberName);
     138. Integer memberIDInterger = (Integer)cache.retrieve(key);
     ...
     144. memberID = memberIDInterger.intValue();
     ...
     154. return memberID;
```



```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
     ...
     94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
     ...
     104. MemberBean memberBean = DAOFactory.getMemberDAO().getMember(memberID);
```



```
175. public MemberBean getMember(int memberID) throws ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
     176. return memberDAO.getMember(memberID);
```



```
1207. public MemberBean getMember(int memberID) //memberdaoimpljdbc.java
     ...
     1220. statement.setInt(1, memberID);
     1221. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 131:

Query Name - **Parameter_Tampering**

Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
     ...
     114. String ret = request.getParameter(param);
     ...
     116. ret = ret.trim();
     ...
     122. return ret;
```



```
125. public void processLogin(HttpServletRequest request, HttpServletResponse
response) //onlineusermanager.java
     ...
     130. String memberName = ParamUtil.getParameter(request, "MemberName", true);
     ...
     140. processLogin(request, response, memberName, memberPassword,
     memberPasswordMD5);
```



```
146. public void processLogin(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
```

```
147. String memberName, String memberPassword, String memberPasswordMD5)
...
166. user = login(request, response, memberName, memberPassword, false);
```

```
193. protected OnlineUser login(HttpServletRequest request, HttpServletResponse
response, //onlineusermanager.java
194. String memberName, String memberPassword, boolean passwordEncoded)
...
211. user = ManagerFactory.getOnlineUserFactory().getAuthenticatedUser
(request, response, memberName, encodedPassword, true);
```

```
68. public OnlineUser getAuthenticatedUser(HttpServletRequest
request, //onlineuserfactoryimpl.java
...
70. String loginName, String password,
...
76. return getAuthenticatedUser(req, null, loginName, password,
isEncodedPassword);
```

```
79. public OnlineUser getAuthenticatedUser(GenericRequest
request, //onlineuserfactoryimpl.java
...
81. String loginName, String password,
...
94. memberID = MemberCache.getInstance().getMemberIDFromMemberName(loginName);
```

```
132. public int getMemberIDFromMemberName(String memberName) //membercache.java
...
141. memberID = DAOFactory.getMemberDAO().getMemberIDFromMemberName
(memberName);
```

```
190. public int getMemberIDFromMemberName(String memberName) throws
ObjectNotFoundException,
DatabaseException //memberdaoexternaluserdatabaseabstract.java
191. return memberDAO.getMemberIDFromMemberName(memberName);
```

```
1406. public final int getMemberIDFromMemberName(String
memberName) //memberdaoimpljdbc.java
...
1416. statement.setString(1, memberName);
1417. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 132:

Query Name - **Parameter_Tampering**

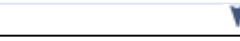
Severity - **Serious**

```
111. public static String getParameter(HttpServletRequest request, String param,
boolean checkEmpty) //paramutil.java
...
```

```
114. String ret = request.getParameter(param);
...
116. ret = ret.trim();
...
122. return ret;
```

```
144. public static int getParameterInt(HttpServletRequest request, String
param) //paramutil.java
...
147. String inputStr = getParameter(request, param, true);
...
150. ret = Integer.parseInt(inputStr);
...
156. return ret;
```

```
396. public void downloadAttachment(HttpServletRequest request, HttpServletResponse
response) //pmattachmentwebhandler.java
...
414. int messageID = ParamUtil.getParameterInt(request, "message");
...
426. messageBean = DAOFactory.getMessageDAO().getMessage(messageID);
```



```
573. public MessageBean getMessage(int messageID) //messagedaoimpljdbc.java
...
586. statement.setInt(1, messageID);
587. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

SQL Injection

CWE ID	89
Description	SQL injection attacks are another instantiation of injection attack, in which SQL commands are injected into data-plane input in order to effect the execution of predefined SQL commands.
Likelihood of Exploit	Very High
Common Consequences	Confidentiality: Since SQL databases generally hold sensitive data, loss of confidentiality is a frequent problem with SQL injection vulnerabilities. Authentication: If poor SQL commands are used to check user names and passwords, it may be possible to connect to a system as another user with no previous knowledge of the password. Authorization: If authorization information is held in a SQL database, it may be possible to change this information through the successful exploitation of a SQL injection vulnerability. Integrity: Just as it may be possible to read sensitive information, it is also possible to make changes or even

delete this information with a SQL injection attack.

Potential Mitigations

Requirements specification: A non-SQL style database which is not subject to this flaw may be chosen.

Design: Follow the principle of least privilege when creating user accounts to a SQL database. Users should only have the minimum privileges necessary to use their account. If the requirements of the system indicate that a user can read and modify their own data, then limit their privileges so they cannot read/write others' data.

Design: Duplicate any filtering done on the client-side on the server side.

Implementation: Implement SQL strings using prepared statements that bind variables. Prepared statements that do not bind variables can be vulnerable to attack.

Implementation: Use vigorous white-list style checking on any user input that may be used in a SQL command. Rather than escape meta-characters, it is safest to disallow them entirely. Reason: Later use of data that have been entered in the database may neglect to escape meta-characters before use. Narrowly define the set of safe characters based on the expected value of the parameter in the request.

Demonstrative Examples

The following code dynamically constructs and executes a SQL query that searches for items matching a specified name. The query restricts the items displayed to those where owner matches the user name of the currently-authenticated user.

C# Example:

```
...
string userName = ctx.getAuthenticatedUserName();
string query = "SELECT * FROM items WHERE owner = '"
+ userName + "' AND itemname = '" + ItemName.Text +
"'";
sda = new SqlDataAdapter(query, conn);
DataTable dt = new DataTable();
sda.Fill(dt);
...
```

The query that this code intends to execute follows: SELECT * FROM items WHERE owner = <userName> AND itemname = <itemName>; However, because the query is constructed dynamically by concatenating a constant base query string and a user input string, the query only behaves correctly if itemName does not contain a single-quote character. If an attacker with the user name wiley enters the string "name' OR 'a'='a" for itemName, then the

query becomes the following: SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name' OR 'a'='a'; The addition of the OR 'a'='a' condition causes the where clause to always evaluate to true, so the query becomes logically equivalent to the much simpler query: SELECT * FROM items; This simplification of the query allows the attacker to bypass the requirement that the query only return items owned by the authenticated user; the query now returns all entries stored in the items table, regardless of their specified owner.

This example examines the effects of a different malicious value passed to the query constructed and executed in the above example. If an attacker with the user name hacker enters the string "hacker'); DELETE FROM items; --" for itemName, then the query becomes the following two queries:

SQL Example:

```
SELECT * FROM items WHERE owner = 'wiley' AND
itemname = 'name';
DELETE FROM items;
--'
```

Many database servers, including Microsoft(R) SQL Server 2000, allow multiple SQL statements separated by semicolons to be executed at once. While this attack string results in an error on Oracle and other database servers that do not allow the batch-execution of statements separated by semicolons, on databases that do allow batch execution, this type of attack allows the attacker to execute arbitrary commands against the database. Notice the trailing pair of hyphens (--), which specifies to most database servers that the remainder of the statement is to be treated as a comment and not executed [19]. In this case the comment character serves to remove the trailing single-quote left over from the modified query. On a database where comments are not allowed to be used in this way, the general attack could still be made effective using a trick similar to the one shown in Example 1. If an attacker enters the string "name'); DELETE FROM items; SELECT * FROM items WHERE owner = 'wiley' AND itemname = 'name'; DELETE FROM items; SELECT * FROM items WHERE 'a'='a'; One traditional approach to preventing SQL injection attacks is to handle them as an input validation problem and either accept only characters from a whitelist of safe values or identify and escape a blacklist of potentially malicious values. Whitelisting can be a very effective means of enforcing strict input validation

rules, but parameterized SQL statements require less maintenance and can offer more guarantees with respect to security. As is almost always the case, blacklisting is riddled with loopholes that make it ineffective at preventing SQL injection attacks. For example, attackers can: - Target fields that are not quoted - Find ways to bypass the need for certain escaped meta-characters - Use stored procedures to hide the injected meta-characters Manually escaping characters in input to SQL queries can help, but it will not make your application secure from SQL injection attacks. Another solution commonly proposed for dealing with SQL injection attacks is to use stored procedures. Although stored procedures prevent some types of SQL injection attacks, they fail to protect against many others. For example, the following PL/SQL procedure is vulnerable to the same SQL injection attack shown in the first example. procedure get_item (itm_cv IN OUT ItmCurTyp, usr in varchar2, itm in varchar2) is open itm_cv for ' SELECT * FROM items WHERE ' || 'owner = '|| usr || ' AND itemname = ' || itm || '; end get_item; Stored procedures typically help prevent SQL injection attacks by limiting the types of statements that can be passed to their parameters. However, there are many ways around the limitations and many interesting statements that can still be passed to stored procedures. Again, stored procedures can prevent some exploits, but they will not make your application secure against SQL injection attacks.

MS SQL has a built in function that enables shell command execution. An SQL injection in such a context could be disastrous. For example, a query of the form:

SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='$user_input' ORDER BY PRICE

Where $user_input is taken from the user and unfiltered.

If the user provides the string:

' exec master..xp_cmdshell 'vol' --

The query will take the following form: "

SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' exec master..xp_cmdshell 'vol' --' ORDER BY PRICE

Now, this query can be broken down into: [1] a first SQL query: SELECT ITEM,PRICE FROM PRODUCT WHERE ITEM_CATEGORY='' [2] a second SQL query, which executes a shell command: exec master..xp_cmdshell 'vol' [3] an MS SQL comment: --' ORDER BY PRICE As can be seen, the malicious input changes the semantics of the query into a query, a shell command execution and a comment.

Observed Examples

CVE-2004-0366 -

CVE-2004-0343 -

CVE-2003-0779 -

CVE-2003-0500 -

CVE-2003-0377 -

Context Notes

SQL injection has become a common issue with database-driven web sites. The flaw is easily detected, and easily exploited, and as such, any site or software package with even a minimal user base is likely to be subject to an attempted attack of this kind. Essentially, the attack is accomplished by placing a meta character into data input to then place SQL commands in the control plane, which did not exist there before. This flaw depends on the fact that SQL makes no real distinction between the control and data planes.

If successful, SQL Injection attacks can give an attacker access to backend database contents, the ability to remotely execute system commands, or in some circumstances the means to take control of the Windows server hosting the database.

Dynamically generating queries that include user input can lead to SQL injection attacks. An attacker can insert SQL commands or modifiers in the user input that can cause the query to behave in an unsafe manner.

Constructing a dynamic SQL statement with user input may allow an attacker to modify the statement's meaning or to execute arbitrary SQL commands.

Factors: resultant to special character mismanagement, MAID, or blacklist/whitelist problems. Can be primary to authentication errors.

References

M. Howard and D. LeBlanc. "Writing Secure Code". 2nd Edition. Microsoft. 2003.

Node Relationships

ChildOf - Injection (74)
ChildOf - Weaknesses in OWASP Top Ten (629)
ChildOf - Weaknesses Examined by SAMATE (630)
ChildOf - Weaknesses Used by NVD (635)
ParentOf - SQL Injection: Hibernate (564)

Source Taxonomies

PLOVER - SQL injection

7 Pernicious Kingdoms - SQL Injection

CLASP - SQL injection

Applicable Platforms All

Path 1:

Query Name - **Blind_SQL_Injections**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
      ...
      642. String order = GenericParamUtil.getParameter(request, "order");
      ...
      671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
      (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```

```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1468. return getBeans_withSortSupport_limit_mysql(offset, rowsToReturn, sort,
      order, memberStatus);
```

```
1495. private Collection getBeans_withSortSupport_limit_mysql(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
      ...
      1541. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
      1542. sql.append(" LIMIT ?, ?");
      ...
      1546. statement = connection.prepareStatement(sql.toString());
      ...
      1549. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 2:

Query Name - **Blind_SQL_Injections**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
      ...
      59. String ret = request.getParameter(param);
      ...
      61. return ret.trim();
```

```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    642. String order = GenericParamUtil.getParameter(request, "order");
    ...
    671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
    (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1470. return getBeans_withSortSupport_limit_noscroll(offset, rowsToReturn,
    sort, order, memberStatus);
```



```
1627. private Collection getBeans_withSortSupport_limit_noscroll(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1672. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
    ...
    1676. statement = connection.prepareStatement(sql.toString());
    ...
    1678. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Path 3:

Query Name - **Blind_SQL_Injections**

Severity - **Serious**

```
57. public static String getParameter(GenericRequest request, String
param) //genericparamutil.java
    ...
    59. String ret = request.getParameter(param);
    ...
    61. return ret.trim();
```



```
631. public void prepareShowUserManagement(GenericRequest
request) //memberwebhandler.java
    ...
    642. String order = GenericParamUtil.getParameter(request, "order");
    ...
    671. memberBeans = DAOFactory.getMemberDAO().getMembers_withSortSupport_limit
    (offset, postsPerPage, sort, order, MemberDAO.ALL_MEMBER_STATUS);
```



```
1465. public Collection getMembers_withSortSupport_limit(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
    ...
    1472. return getBeans_withSortSupport_limit_general(offset, rowsToReturn,
    sort, order, memberStatus);
```



```
1760. private Collection getBeans_withSortSupport_limit_general(int offset, int
rowsToReturn, String sort, String order, int memberStatus) //memberdaoimpljdbc.java
```

```
...
1807. sql.append(" ORDER BY " + sort + " " + order);// ColumnName, ASC|DESC
...
1811. statement = connection.prepareStatement(sql.toString(),
ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
...
1818. resultSet = statement.executeQuery();
```

[Table of Contents] [Description]

Data Leaking Between Users

CWE ID	488
Description	Data can "bleed" from one session to another through member variables of singleton objects, such as Servlets, and objects from a shared pool.
Demonstrative Examples	The following Servlet stores the value of a request parameter in a member field and then later echoes the parameter value to the response output stream.

```
public class GuestBook extends HttpServlet {

  String name;

  protected void doPost (HttpServletRequest req,
  HttpServletResponse res) {
    name = req.getParameter("name");
    ...
    out.println(name + ", thanks for visiting!");
  }
}
```

While this code will work perfectly in a single-user environment, if two users access the Servlet at approximately the same time, it is possible for the two request handler threads to interleave in the following way: Thread 1: assign "Dick" to name Thread 2: assign "Jane" to name Thread 1: print "Jane, thanks for visiting!" Thread 2: print "Jane, thanks for visiting!" Thereby showing the first user the second user's name.

Context Notes	Many Servlet developers do not understand that, unless a Servlet implements the SingleThreadModel interface, the Servlet is a singleton; there is only one instance of the Servlet, and that single instance is used and re-used to handle multiple requests that are processed simultaneously by different threads. A common result of this misunderstanding is that developers use Servlet member fields in such a way that one user may inadvertently see another user's data. In other words, storing user data in Servlet member fields introduces a data access race condition.
Node Relationships	ChildOf - Encapsulation (485)

Source Taxonomies 7 Pernicious Kingdoms - Data Leaking Between Users
Applicable Platforms All

Path 1:

Query Name - **Singleton_HTTPServlet**

Severity - **Serious**

```
65. public class ForumAdminServlet extends HttpServlet //forumadminservlet.java
```

[Table of Contents] [Description]

Path 2:

Query Name - **Singleton_HTTPServlet**

Severity - **Serious**

```
64. public class ForumUserServlet extends HttpServlet //forumuserservlet.java
```

[Table of Contents] [Description]

Copyright © 2008 Checkmarx
Description materials copyrighted by the National Cyber Security Division of the U.S. Department of Homeland Security and the MITRE Corporation.