MBE 2010
Video Track
Guideline for Participants

21 January 2010

Table of Content

1 Introduction

This document specifies the rules and guidelines for executables that are submitted to the MBE 2010 Video Track. The MBE 2010 Video Track's goal is to assess the performance of face recognition from video. This includes matching still images to video and video to video. Section 2 discusses the MBE 2010 Video Track test environment. MBE 2010 Video Track participants must ensure that their applications are compliant with these requirements. Instructions for creating executables, including naming convention, calling signatures and arguments, and packing applications, are described in Section 3. Format specifications for MBE 2010 Video Track files, such as parameters files, signature set (*Sigset*) and similarity matrices, are described in Section 4. Section 5, answers Frequently Asked Questions. Topics not covered in this document should be directed to the MBE 2010 Liaison at mbe2010@nist.gov.

2 MBE 2010 Video Track Test Environment

2.1 Computers Configuration

The intended hardware for the MBE 2010 Video Track is Intel-based servers. All submitted executables will be transferred to the MBE 2010 Video Track servers and executed using the directory structures and protocol described below. The MBE 2010 Video Track computing environment puts the following requirements on your executable:

- A separate executable must be submitted for each experiment type.
- Executables must be able to be installed multiple times on (potentially) multiple servers.
- Executables must run on a stand-alone machine that does not have internet access during the MBE 2010 Video Track evaluations. Please ensure that your executables can be installed and executed without access to the internet.
- All libraries (static or dynamic) used for the executable must be included in the application package. DO NOT ASSUME THAT ANY STATIC OR DYNAMIC LIBRARIES EXIST ON THE MBE 2010 VIDEO TRACK SYSTEMS. It is recommended that you verify that you are not using any non-standard libraries by wiping a system clean (i.e. reformatting the disk drive), reinstalling the operating system and then verifying that your executable runs with no additional libraries.

2.2 Operating Systems

The MBE 2010 Video Track servers will have either the Windows XP Professional or the Linux operating system (OS). Both OS will be 32-bit variant. The targeted versions of the operating systems are Windows XP Professional with Service Pack 3 and Fedora 12.

2.3 Directory Structure of Application

Each MBE 2010 Video Track participant must utilize the directory structure shown in Figure 1. This directory structure will be repeated for each installation of the executable. It is important that participants make no assumptions as to the disk drive or parent directory that this directory structure is installed in. Thus, participants should not assume that the top-level directory is on the *C:\* drive or in */home/*.

The name of the top level directory must correspond to the name of the executable as specified in *Section 3.1*. All other directories will be below the top level directory. MBE 2010 Video Track participants will not have access privileges to any directory outside of this directory hierarchy. For example, participants will NOT be able to write temporary files to *C:\temp*, */tmp/* or */usr/tmp*.

- Executables must be installed to the *executable_name/bin/* directory.
- All experiments (and thus executables) will be instantiated from the top level (*executable_name/*) directory. Thus, a typical calling syntax will be of the following form:

  bin/executable_name <argument_list>

- All scripts, data, etc. that an executable needs should be stored in the *executable_name/bin/* directory structure.
- All libraries needed by executables should be installed in either the *executable_name/bin/* or the *executable_name/lib/* directory.
- After installation, both the *executable_name/bin/* and the *executable_name/lib/* directories will be made read and execute only. If a subdirectory structure is needed, then it should be created by the installation process. After the executable installation process is complete, the *executable_name/bin/* and *executable_name/lib/* directory structures cannot be modified.

- All required outputs (i.e. similarity matrices) generated during experiment execution will be written to the *executable_name/output* directory that is specified in the executable calling signature. Typically, this will be the *executable_name/output/* directory. However, participants should not hardcode the name of the output directory rather they should write their output to the file specified in the calling signature. Note, output files names may be specified as either absolution filenames (e.g. c:/MBE/MyExecutable/output/sim_matrix.mtx) or relative filenames (e.g. output/sim_matrix.mtx). Regardless of how it is specified, users will be given write permission to the *executable_name/output/* and/or appropriate directory.

- Partial and/or temporary results can be stored in the *executable_name/temp/* directory. Users will be will be given both read and write privileges to the *executable_name/*/temp/ directory given. It is also permissible for them to create and populate other subdirectories within the top-level *executable_name/temp/* directory. At the completion of an experimental run, the contents of the executable_name/temp/ directory structure maybe erased at the MBE 2010 evaluation team's discretion. Alternatively, the team may decide to reuse biometric samples during a sequence of experimental runs. To expedite processing, templates should be stored in the *executable_name/temp/* directory structure and reused where applicable. However, the code should be designed to handle cases were the templates are erased prior to the experiment. Partial results can also be stored in the *executable_name/temp/* directory. It is important to note that files output to the temporary directory may be deleted prior to subsequent runs of experiments.

- Images will be stored outside of the directory structure shown in Figure 1. Users will be given either the absolute or the relative path to the images via the *image_directory* parameter (see Section 0).

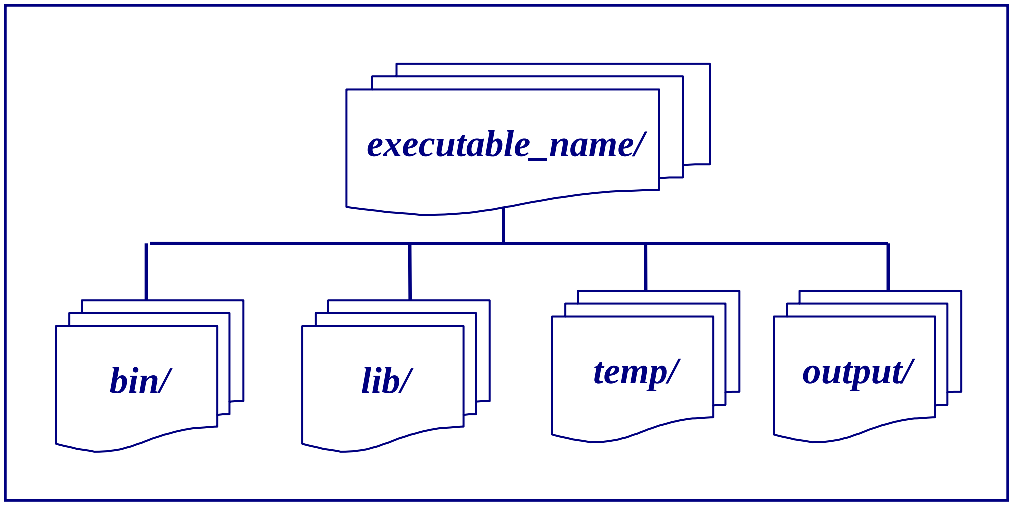

Figure 1 MBE 2010 Video Track Directory Structure

3 Creating Executables

3.1 *1-1* versus *1-Many* Executables

MBE 2010 Video Track will evaluate both *1-1* and *1-many* executables. *1-1* (or verification) executables must produce the same score when comparing sample *i* to sample *j* independent of the size and/or composition of the target and query sets. *1-many* (or identification) algorithms can normalize scores based on the composition of the target set. For *1-many* tests, we will ensure that each signature (subject) is only in the target set at most one time.

MBE 2010 Video Track will enforce a set of independence requirements for *1-1* submittals. These requirements ensure that submitted *1-1* executables produce the same scores in batch mode that they produce when executed in a one-to-one mode. To this end, a set of validation tests will be used to confirm the independence of each *1-1* executable. Each submitted *1-1* executable must pass these tests to confirm that it satisfies the independence requirements.

MBE 2010 Video Track participants must submit at least one (validated) *1-1* executable, Submission of *1-many* executables are optional.

Please review the naming convention in the next section to properly denote the intended matching type of your submittal.

3.2 Robustness to Image Quality

Submitted executables must be able to process biometric samples (still images and video sequences) of varying quality without failure. More specifically, all executables must create a similarity score for all specified comparisons – even if the executable fails to create a template for the biometric sample. Incomplete similarity matrices will cause inaccurate accuracy scores to be calculated during the evaluation. It is recommended that a descriptive message be written to the log file when an executable has problems with a biometric sample.

3.3 Executable Naming Convention

This section specifies the naming conventions for MBE 2010 Video Track submittals. The name of all MBE 2010 Video Track executables are consistent in that they consist of a base name, followed by an underscore, followed by the experiment name, followed by an underscore, followed by the matching type, followed by an underscore, followed by the algorithm version. The typical naming syntax is:

basename_experimentType_matchingType_algorithmVersion

For example:

NIST_VV-VV_1-N_1

Substrings	Allowed Values	Comment
basename	(to be emailed)	Unique name given to each participant via an e-mail at registration.
experimentType	VV-VV	VV-VV: Visible Video to Visible Video
matchingType	1-1 1-N	1 to 1 (verification) or 1 to Many (identification)
algorithmVersion	1 or 2	The first submittal of an algorithm should be labeled 1 the second 2 (if applicable).

Table 1 Executable Naming Convention

- The base name, experiment name, matching type and the algorithm version are separated by underscores.

- Base names will be unique to each participant. Participants will receive an e-mail from the MBE 2010 Liaison that specifies the base name that they should use with all of their executables and corresponding top-level directories.
- The experiment type corresponds to the modalities to which the algorithm is applicable. The only experiment type described in the video track is visible video to visible video which is denoted “VV-VV”. The VV-VV experiment has two components: Still Image to Video matching and Video to Video matching. All executable submissions must be able to handle both of these components.
- Participants can submit up to two executables representing different algorithms for a given experiment type/matching type combination. If a second executable is submitted, then an explanation should be provided that explains the differences between the executables (e.g. submittal 1 is optimized for high-resolution images while submittal 2 is optimized for low resolution/quality images). The version of multiple algorithms must be indicated in the executable name as described above. Please note that we recommend that participants label the algorithm version of their primary submittal as “_1”.

3.3.1 1-1 Matching Executables

Experiment	Executable Name
Visible Video to Visible Video	*basename*_VV-VV_1-1_*algorithmVersion*

3.3.2 1-many Matching Executables

Experiment	Executable Name
Visible Video to Visible Video	*basename*_VV-VV_1-N_*algorithmVersion*

3.4 Executable Calling Signatures & Arguments

3.4.1 Calling Signatures

All MBE 2010 Video Track executables will be called from the command line. The calling signature for MBE 2010 Video Track executables is:

executableName* *parameter_file* *image_directory* *target_sigset* *query_sigset* *similarity_file

3.4.2 Arguments

Table 2 describes the arguments used above in the calling signatures for the executables. It is important to note that all filenames will be relative to the working directory from which the program is instantiated.

Parameter name	Type	Format	Description
parameter_file	Input	XML	An XML document that specifies experiment description information, configuration parameters and the name of metadata files.
image_directory	Input	string	The relative path to the image (data) directory
target_sigset	Input	Sigset	The name of the target signature set. This document will contain a list of the target images.
query_sigset	Input	Sigset	The name of the target signature set. This document will contain a list of the query images.
similarity_file	Input/ Output	Similarity Matrix	The name of the similarity file. This will be the primary output data structure.

Table 2 Descriptions of Executable Arguments

Remarks

- A file name may be specified with either an absolution path (e.g. C:\MBE\MyExecutable\output\similarity.mtx) or a relative path (e.g. output\similarity.mtx) from the current directory.
- Filenames and directory names will not contain whitespace characters. We will use an underscore or an uppercase letter to separate words in filenames.

3.4.3 Image Directory

The image directory is a string that specifies the absolute or relative path to the directory that contains images. This string should be appended to each of the data files listed in the signature sets. For examples, if the parameter *image_directory* is '..\images\' and the sigset refers to an image named '2D_Visible\0023d1.mpg', you should process the image in the file '..\images\2D_Visible\0023d1.jpg'. Please note that the value of the *image_directory* argument will be consistent with the underlying operating system. Thus, '..\images\' and '../images/' would be provided for Windows and Linux operating system respectively.

3.5 Packaging & Installing Your Application

Executables can be submitted on a CD, DVD or USB stick. Executables should be packaged to ensure that the MBE 2010 evaluation team can easily and correctly install your submittal on the MBE 2010 Video Track test systems. Two types of installation processes are acceptable. Your submittal must have either an installation program that performs all files extractions and configurations or your submittal must be installable via direct copy of the content of your disk (CD, DVD, or USB) to the corresponding directory on the target platforms. In the latter case, installation should not require any additional configuration steps In any event; the MBE 2010 evaluation team will ensure that the directories have the appropriate access privileges as detailed in Section 2.3.

3.5.1 Installation Utility Method

You must provide an installation program (or script) if setup and configuration of your submittals requires steps other than simple copying of files and setting of directory permissions. In this case, each executable must have its own installation program (or script). The top level directories on the disk (CD, DVD, or USB) must correspond to the names of the executables as specified in Section 3.1 of this document. Inside of each top level directory, there should be a clearly named install executable file (e.g. install.bat, install.exe, install.sh, etc). Your install program must be executable directly from your disk (CD, DVD, or USB) (i.e. the install program copies all necessary files to a user supplied directory) or it must be executed after the entire content of the disk (CD, DVD, or USB) is copied to the target system. There should be a readme.txt file in each directory that specifies whether the install program should be run directly from the disk (CD, DVD, or USB) or whether the files should be copied to the home directory prior to running the install program. Upon execution, this install file should copy all necessary files (executables, libraries, parameter files) to the appropriate directory on the target platform. The install file should also perform any necessary configuration. Other than execution of the install file and setting of directory permissions, no additional setup steps should be necessary to run the executables on the target system.

3.5.2 Copy Directory Method

In this method, participants must place all files necessary to execute their submittals in a directory structure that mimics the directory structure specified in Section 2.3. Specifically, the top level directories on the disk (CD, DVD, or USB) must correspond to the names of the executables as specified in Section 3.1 of this document. There should be a *bin/* directory and a *lib/* directory directly beneath each top level directory. The content of the *bin/* and *lib/* directories will be copied to the corresponding directory on the target platform. No additional setup (other than setting of directory permissions) should be necessary to run the executables on the target system. There should also be a *readme.txt* in each directory that describes the contents of the directory.

4 MBE Video Track File Formats

All MBE 2010 Video Track submittals will be required to process several standard file formats. The use of standard formats is necessary to enable systematic test and evaluation of the executables on a common set of images using validated test protocols.

4.1 Parameter Files

4.1.1 Overview

Parameter files are XML documents that provide experiment description information and configuration values for executables. To simplify their processing, configuration values will always be specified via the values of attributes in the parameter file.

4.1.2 Structure

While the precise content of the Parameter files has not been determined, the structure will be similar to the example shown below in **Figure 2**. In this example, elements are depicted blue (capitalized), attributes are red (lowercase) and attribute values are black (uppercase and in quotes). The *Experiment* element is the outer element. It has one attribute named *type* with value "1-many". The *Experiment* element has three child elements *Target, Query* and *LogFile*. Both the *Target* and *Query* elements have four attributes: spectrum, *pose*, *min_recordings* and *max_recordings*. The *LogFile* element have on attribute named *name*.

```
<?xml version="1.0"?>
<Experiment name="Out_Activity" type="1-many" >
 <Target spectrum="Visible" min_recordings="1" max_recordings="1" pose="Varied" />
 <Query spectrum="Visible" min_recordings="1" max_recordings="1" pose="Varied" />
 <LogFile name="1_to_many_logfile.txt" />
</Experiment>
```

Figure 2 Example of a Parameter File

The parameters and their allowed values are described below in Table 3.

Attribute	Type	Allowed Values	Comments
name (Experiment)	String		The text of this string provides a name for the experiment. This attribute is not needed by users.
type	String	1-1 1-many	The value of this attribute specifies whether the algorithm should perform *1-1* (verification) of *1-many* (identification) matching. Identification algorithms can optionally use score normalization. Note: *1-1* algorithms can be used on both *1-1* and *1-many* experiments while *1-many* algorithms can only be used on *1-many* experiments.
spectrum	enumerated	Visible	The enumerated values of this attribute specify the electromagnetic spectrum of the images. The only spectrum used in the Video Track is EOI visible.
capture	enumerated	Controlled Uncontrolled	The enumerated values of this attribute specify whether the images were captured under controlled or uncontrolled conditions.
min_recordings	integer	>0	This integer specifies the minimum number of images that will be associated with a signature. This value will be 1 for all experiments in the video track since all signatures will have precisely one image.
max_recordings	integer	>0	This integer specifies the maximum number of images that will be associated with a signature. This value will be 1 for all experiments in the video track since all signatures will have precisely one image.

Attribute	Type	Allowed Values	Comments
pose	string	Frontal Nonfrontal Varied	Pose variation of the biometric signatures in the sig-set.
name (LogFile)	string	Filename	All logging information generated by the program should be written to a file with this name in the output directory.

Table 3 Description of Elements a Parameter File

Please note that for the MBE 2010 Video Track the spectrum will always be visible, and for the video sequences the max-recordings will be one.

4.1.3 Parsing

Due to their simple structure, Parameter files are readily parsed with any XML or XPath parser. Source implementation of C++ and Java classes for parsing Parameter files are provided in the BEE (Biometric Experimentation Environment) distribution. These classes, which use the XPath parser, are available to MBE 2010 Video Track participants. Parsers written in other languages (e.g. C#) will not be made available. However, users should be able to easily create parsers in other languages using the C++ classes as a guide.

4.1.4 Writing

MBE 2010 Video Track participants will not be required to write (output) Parameter files.

4.2 Log Files

Each executable should produce a log file. The name (and directory) for the log file will be provided in the parameter file via the *name* attribute of the *LogFile* element. The log file should provide sufficient information so that the MBE 2010 evaluation team (with limited support from the submitter) can easily and quickly troubleshoot submitted executables. Please note that the MBE 2010 evaluation team we will have limited time to troubleshoot individual submittals because of the large number of submittals and scheduling. This is it important that each submittal provide detailed debug information in the log file. For example, consider the case where the algorithm is unable to find the image specified in the signature set. A useful log entry will specify both the error type and the full name of the image (i.e. "*** Error – unable to open ..\MBE\images\IR-video\00056d123.ts ***").

4.3 Signature Sets (*Sigsets*)

4.3.1 Overview

Signature Sets (Sigsets) are the primary input structure for the MBE 2010 Video Track. They will be used to specify the files in the Target and/or Query sets.

4.3.2 Structure

The Signature Set document will provide a list of images. XML will be used because its hierarchical structure facilitates a flexible representation of the relationships between subjects, sessions, sensors and files. Specifically, the Signature Set will consist of a list of *Signature* (subjects) element. Each *Signature* element will contain one or more *Presentation* child elements that correspond to capture sessions. Each *Presentation* element will contain one or more *Component* elements that correspond to a sensor. Lastly, each *Component* will have one or more *Data* elements that correspond to a file.

Figure 3 and Figure 4 illustrate the general Signature Set structure and the simple Siguature Set structure, respectively, with elements depicted blue, attributes depicted red (lowercase) and attribute values depicted black (uppercase and in quotes).

The first example in Figure 3 has three **biometric-signature** elements and represents three subjects. The signatures are named "S5288", "S5156", and "S5304". Each signature has one **presentation**, and each **presentation** has four attributes: **name**, **modality**, **file-name**, **and file-format**. For instance, the signature "S5288" has a presentation labeled "S5288", same as the signature name. And the presentation consists of a visible video sample stored in the MPEG-TS file "vf/5288d1.ts".

```xml
<?xml version="1.0" encoding="UTF-8"?>
<biometric-signature-set  xmlns="http://www.bee-biometrics.org/schemas/sigset/0.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.bee-biometrics.org/schemas/sigset/0.1
    http://www.bee-biometrics.org/schemas/sigset/0.1/simple.xsd">
  <biometric-signature name="S5288" >
    <presentation name="S5288" modality="Face" file-name="vf/5288d1.ts" file-format="MPEG-TS" />
  </biometric-signature>
  <biometric-signature name ="S5156" >
    <presentation name ="S5156" modality="Face" file-name="vf/5156d3.ts" file-format="MPEG-TS"/>
  </biometric-signature>
  <biometric-signature name ="S5304" >
    <presentation name ="S5304" modality="Face" file-name="vf/5304d7.ts" file-format="MPEG-TS"/>
  </biometric-signature>
</biometric-signature-set>
```

Figure 3 Example of a Simple Signature Set

The second example in Figure 4 has two **complex-biometric-signature** elements and thus represents two subjects. The **complex-biometric-signature** are named "**nd1S04327**" and "**nd1S04385**". Both **complex-biometric-signatures** consist of a single **complex-presentation** that represents a single visit (session) of the subject. The **complex-presentations** have the same label as the parent **complex-biometric-signature**. (Note: While **complex-presentations** grouped under a single **complex-biometric-signature** are guaranteed to correspond to the same subject, separate **complex-biometric-signatures** can also correspond to the same subject.) Both **complex-presentations** have two **presentation-component**s (separate sensors). The **presentation-components** are labeled **"Face_HD"** and **"Face_NIR"**. The **presentation-components** consists of a single HD face video sample stored in the MPEG-TS file format and a NIR face video stored in the AVI file format. The files in **complex-presentation** **"nd1S04327"**are named **"FaceVisibleVideo/04327v1134.ts"** and **"FaceNIRVideo/04327v1133.avi"** respectively.

```
<?xml version="1.0" encoding="UTF-8"?>
<biometric-signature-set  xmlns="http://www.bee-biometrics.org/schemas/sigset/0.1"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.bee-biometrics.org/schemas/sigset/0.1
  http://www.bee-biometrics.org/schemas/sigset/0.1/general.xsd">
 <complex-biometric-signature name="nd1S04327" >
  <complex-presentation name="nd1S04327" modality="Face" >
   <presentation-component name="Face_HD" >
    <data file-name="FaceVisibleVideo/04327v1134.ts" file-format="MPEG2-TS" />
   </presentation-component>
  </complex-presentation>
  <complex-presentation name="nd1S04327" modality="Face_NIR" >
   <presentation-component name="Face_NIR" >
    <data file-name="FaceNIRVideo/04327v1133.avi" file-format="avi" />
   </presentation-component>
  </complex-presentation>
 </complex-biometric-signature>
 <complex-biometric-signature name="nd1S04385" >
  <complex-presentation name="nd1S04385" modality="Face" >
   <presentation-component name="Face_HD" >
    <data file-name="FaceVisibleVideo/04385v1302.ts" file-format="MPEG2-TS" />
   </presentation-component>
  </complex-presentation>
  <complex-presentation name="nd1S04385" modality="Face_NIR" >
   <presentation-component name="Face_NIR" >
    <data file-name="FaceNIRVideo/04385v1299.avi" file-format="avi" />
   </presentation-component>
  </complex-presentation>
 </complex-biometric-signature>
</biometric-signature-set>
```

Figure 4 Example of a Complex Signature Set

4.3.3 Parsing

Signature sets are difficult to parse due to their rich hierarchical structure. Fortunately, C++ and Java classes for parsing Similarity Matrices are provided in the BEE (Biometric Experimentation Environment) distribution. These classes are available to MBE 2010 Video Track participants. We also provide examples for the use of these parsers. It is important to note that some of the BEE examples assume the simplified Sigset structure in which each Signature has precisely one Presentation, each Presentation has precisely one Component and each Component has precisely one Data member. Care should be used which using this simplified version of the Similarity structure. Parsers are not provided for other languages (e.g. C#). However, users should be able to create a Java wrapper that parses the Signature Set and passed the appropriate data structures to this executables.

4.3.4 Writing

Like most XML documents, Signature sets are easier to write than they are to parse. Thus, users can either use the C++ and Java classes supplied in BEE to create Signature Sets or output them directly. We recommend that the supplied classes be used because they have been rigorously tested and can easily be made compliant with changes in the schema for Signature Sets.

4.4 Similarity Matrices

4.4.1 Overview

Similarity Matrices are the primary output structures of recognition algorithms in MBE 2010 Video Track. They consist of a header that specified the type and dimension of the contained data and the $N{\times}M$ scores from the biometric algorithm. It is important to note that executables must create a similarity score for all specified comparisons – even if the executable fails to create a template for the biometric sample.

4.4.2 Structure

The Similarity Matrix is similar to many image files in that it contains a textual header followed by a binary representation of an $N{\times}M$ data structure. The structure of the header is depicted below in Figure 5. Here, we see that the header consist of four lines. The first line must contain the either the character ‘D’ (for distance matrix) or the character ‘S’ (similarity matrix) followed by the character ‘2’. The second and third lines should contain the name of the target and query signature set respectively. The target and query name should be the same as they were specified in the call to the matching executable. The fourth line should contain the characters ‘MF’, a space, the number of signatures in the query sigset, a space, the number of signatures in the target sigset, a space and the integer 0x12345678 written in binary format. All four lines in the header should be terminated by a carriage return (CR) character. Table 4 describes the elements in the similarity header.

```
D2
BEE_DIST/signature_sets/MBE_Exp_1.0_Target.xml
BEE_DIST/signature_sets/MBE_Exp_1.0_Query.xml
MF 1425 1425 xV4
```

Figure 5 Example of the Similarity Matrix Header

Name	Format	Separator	Comments
Storage Type	Character 'S' or 'D'	none	Specified similarity scores, 'S', or distance measures 'D'.
Version	The character '2'	CR[1]	The value '2' corresponds to the version of similarity matrix.
Target name	string	CR (0x0d)	This string should be the same as the name of the target sigset provided to the matching algorithm.
Query name	string	CR (0x0d)	This string should be the same as the name of the query sigset provided to the matching algorithm.
Format	The characters 'MF'	space (0x20)	The values correspond to a matrix, 'M', containing float, 'F', values.
Rows	integer	Space (0x20)	The number of signatures in the query set.
Cols	integer	Space (0x20)	The number of signatures in the target set.
Magic number	0x12345678 (4-bytes binary)	CR (0x0d)	This binary value is used to check for Endian (byte swapping).

Table 4 Description of Elements in the Similarity Header

The scores are written to the file immediately following the header. These should be $N \times M$ 4-byte binary floating point values. Here, N is the number of signatures in the query set and M is the number of signatures in the target set. Thus, the first M values correspond to comparing the first query signature (sample) to each of the target signatures (samples). There must **not** be any white space characters separating scores (or lines) in the body of the Similarity Matrix.

4.4.3 Parsing

C++ and Java classes for parsing Similarity Matrices are provided in the BEE (Biometric Experimentation Environment) distribution. These classes will are available to MBE 2010 Video Track participants. Parsers written in other languages (e.g. C#) will not be made available.

[1] CR: the ASCII carriage return character (0x0d).

However, users should be able to easily create parsers in other language using the C++ classes as guides.

4.4.4 Writing

C++ and Java classes for writing Similarity Matrices are provided in the BEE (Biometric Experimentation Environment) distribution. These classes are available to MBE 2010 Video Track participants. Parsers written in other languages (e.g. C#) will not be made available. However, users should be able to easily write Similarity Matrices in other language using the C++ classes as guides.

4.5 Imagery

4.5.1 Overview

MBE 2010 Video Track participants will be required to process both still frame images and video sequences. Still frame images will be in either the JPEG (.jpg) or the JPEG 2000 (.jp2) format. Video sequences will be in the AVI (.avi), the Digital Video (.dv), or the MPEG-TS (.ts) format.

4.5.2 Structure

The structure of JPEG, JPEG 2000, MPEG-TS, DV and AVI files are published as ISO standard and/or are readily available on the internet. Additionally, information reading these formats is available in ISO specifications.

4.5.3 Parsing

Libraries containing parsers for JPEG, JPEG 2000, AVI, DV and the MPEG formats are readily available as freeware, shareware and commercially.

4.5.4 Writing

MBE 2010 Video Track participants will not be required to write (output) imagery and therefore are not required to create images in the JPEG, JPEG 2000, AVI, DV or MPEG formats.

5 FAQs

Q1: Should I assume that my executable will run in the *bin/* directory?

A1: No. The executable will be run from the top-level directory of our package. The name of the executable will be specified relative to this directory.

Q2: Which libraries (.dll and .lib) should be included in the submission?
A2: All libraries (static and dynamic) that are not part of the underlying operating system and are necessary to run your application must be included in the submission

Q3. How can I ensure my executable is not dependent on any libraries installed on my machine that will not be on the MBE 2010 Video Track systems?
A3. We recommend that prior to submitting your application, you wipe a system clean (e.g. reformat the HDD), install the OS and then install your application and execute it.

Q4: In which directory should DLLs be placed?
A4: DLLs should be placed in either the bin/ directory or in the *lib/.*

Q5: Will the MBE 2010 Video Track filenames and/or directory names contain blanks (or whitespace)?
A5: No. We will not utilize any filenames or directory names that contain whitespace characters. Our naming convention uses an underscore or an uppercase character to separate words within filenames.

Q6: What kind of information should be written to the log file?
*A6: A log file output should include information that will help to determine the cause of a potential interruption of execution. It should be informational enough to indicate why and where the executable stopped. For example, if a file was not found, a good log output will state "*** ERROR – the image file filename_including full_path was not found. ***"*

Q7. What are some of the common reasons that executable failed during previous evaluations?
Q7. Common causes of execution interruption include:

- *Missing libraries necessary to run the executable;*

- *The executable could not locate the specified file;*
- *The executable could not open the specified file;*
- *The executable locks up, crashes or throws an exception while processing a sample (i.e. image or video);*
- *An executable could not create a template from the specified file (e.g. insufficient quality);*
- *The executable has a memory leak and used all available system memory;*
- *The executable tried to access a directory that it does not have read/write permission;*
- *The executable does not accept the specified arguments;*
- *Input or output directories or filenames are hardcoded in the executable and thus are not read properly from the command line.*

Q8: What should my executable do if it cannot create a template for a given biometric sample (image)?
A8: Submitted executables must create a similarity score for all specified comparisons – even if the executable fails to create a template for the biometric sample. Incomplete similarity matrices will cause inaccurate accuracy scores to be calculated during the evaluation.

Q9: Can I submit multiple algorithms?
A9: *Yes. In addition to being able to submit both a 1-1 and a 1-many algorithm, participants can submit up to two executable for a given experiment type/matching type combination. For instance, participants can submit two "VV-VV_1-1" executables and a "VV-VV_1-many" executable. If a second executable is submitted for an experiment type/matching type combination, then an explanation must to be provided describing the reason for multiple executables (e.g. submittal 1 is optimized for high-resolution images while submittal 2 is optimized for low resolution/quality images). The version of multiple algorithms must be indicated in the executable name as described in Section 3.1* (e.g. the *"VV-VV_1-1" executables should be labeled "VV-VV_1-1_1"and "VV-VV_1-1_2" executables respectively). We recommend that participants label the algorithm version of their primary submittal as "_1".*

Q10: What are the experiment types for the video evaluations?
A10: All submittals in the video evaluation must use (and support) the visible video to visible video, "VV-VV" experiment type. The VV-VV experiment has two components: Still Image to Video matching and Video to Video matching. All executable submissions must be able to handle both of these components. Still images consist of a single image stored in the JPEG or JPEG 2000 format. Video sequences in the MBE 2010 Video Track will be in the AVI, DV or MPEG-TS file formats.

Q11: Does the Video Track use the complex signature set structure shown in Figure 4? If not, why was it included?
A11: No, the MBE 2010 Video Track signature sets all adhere to the simple signature set structure. However, the signature set parsers included in the BTools distribution parses both formats seamlessly. We recommended that you implement the more generic parser and use it for both formats. This will minimize the necessary changes to your code if you apply it to other MBE 2010 Tracks.

Q12: What format will the input (raw) images be in?
A12: MBE 2010 Video Track participants will be required to process both still frame images and video sequences. Still frame images will be in either the JPEG (.jpg) or the JPEG 2000 (.jp2) format. Video sequences will be in the AVI (.avi), the Digital Video (.dv) or the MPEG-TS (.ts) format.

Q13: Can I submit a draft version of my executable prior to the final submittal date so that the MBE 2010 evaluation team can confirm compliance with the MBE 2010 Video Track guidelines?
A13: Yes. MBE 2010 Video Track participants may submit their executables two weeks prior the final due day for compliance verification. The sole purpose of pre-submittal is to ensure that the executables followed the rules in this document. The MBE 2010 Video Track evaluations will be performed on the final submittals - not the pre-submittal.