2012 TRECVID MULTIMEDIA EVENT DETECTION EVALUATION PLAN

1   Introduction

This document is the evaluation plan for Multimedia Event Detection (MED) track of the 2012 TRECVID evaluation. The multi-year goal of MED is to support the creation of *ad hoc* event detection technologies that will permit users to define their own complex events and to quickly and accurately search large collections of multimedia clips.

A MED system is defined to have three separate phases:
1) **metadata generation**   (video ingest and metadata store creation),
2) **event agent generation**   (event definition ingest and event agent creation), and
3) **event agent execution**   (search)

A MED event is a complex activity occurring at a specific place and time which involves people interacting with other people and/or object(s). A MED event consists of a number of human actions, processes, and activities that are loosely or tightly organized and that have temporal and semantic relationships to the overarching activity. All MED events are directly observable.

Searches will run one event at a time against the **metadata store**. The metadata store must be created prior to ingesting the event definition (and the search), thus the metadata store must be sufficiently rich such that it could be searched successfully by any *ad hoc* event with limited event exemplars.

Highlights of the MED 2012 evaluation (MED12) include:
- A new data collection containing over ~4000 hours of multimedia clips
- 10 new events for the Pre-Specified event task
- The introduction of the Pilot Ad Hoc event task with 5 new events.

Participants may use any computer system architecture to generate the metadata store. However, participants must use ONLY COTS standard personal computing platform(s) to run searches. The searches are to be performed locally and the output of the searches is to be submitted to NIST as specified below for official scoring and analysis.

The MED evaluation is open to all that find the task of interest and who are willing to abide by the rules of the evaluation.

2   Evaluation Tasks

MED12 supports two evaluation tasks, namely the Pre-Specified event task and Ad Hoc event task.

2.1   The Pre-Specified Task   (required task)

The Pre-Specified (**PS**) task is analogous to the MED11 task and is an intermediate step towards the goal of Ad Hoc event search. For this task the metadata store may be optimized with knowledge of pre-

specified events to be used for evaluation.   Although developers may tune metadata generation to the PS events, the metadata must be frozen prior to event agent execution.

2.2   The Ad Hoc Task   (optional task)

The pilot Ad Hoc (**AH**) task uses the same frozen metadata (store) created to support the **PS** task. Therefore the AH task tests search capabilities when metadata generation was **NOT** optimized for a known set of events.

Both, PS and AH systems are to generate the same system outputs.

3   Evaluation Conditions

MED12 evaluation conditions include event agent processing style, events processed, event training resources, metadata generation speed, and event agent execution speed.

All evaluation conditions are independent system attributes, meaning systems can be built and submitted for any combination of conditions. However, participants are required to make at least one formal submission using the common set of conditions.

3.1   Event Agent Processing Style

The Event Agent Generation (EAG) phase can process the event kits in two modes: automatic (**AutoEAG**) where no human intervention is used to build the event agent, and semi-automatic (**SemiAutoEAG**) where the EAG phase includes human interaction to build the agent.

System runs with either mode are acceptable.

3.2   Events Processed

MED12 will include 20 PS events (10 MED11 events and 10 new events) and 5 AH events. PS systems that process all twenty PS events and AH systems that process all 5 AH events will be referred to as **MEDFull** systems, while systems that process less than the full set of events (for a task) will be referred to as **MEDPart** systems[1]. Systems developed separately for *MEDFull* and *MEDPart* are not comparable and therefore comparison of such results will not be made by NIST.

All participants must minimally build a *MEDPart* system but are encouraged to address the full challenge posed by *MEDFull.*

3.3   Event Training Conditions

Results from the previous MED11 evaluation indicate the number of video exemplars used during EAG affects system performance. MED12 specifies two levels of exemplars used for EAG.

- *EKFull*: The textual event definition and all provided videos are used during EAG.
- *EK10Ex*: The textual event definition and a specified set of 10 positive and 10 related video exemplars are used during EAG.

EKFull is required for all participants. EK10Ex is optional.

---

[1] For the Dry Run, only 10 events are required to be a MEDFull system.

Condition	Pre-Specified Task (required choice)	Pre-Specified Task (discretionary choice)
Events Processed		MEDFull or MEDPart (MEDFull preferred)
Event Agent Generation		AutoEAG or SemiAutoAEG
Event Training	EKFull	
Metadata Generation Speed	MGSU	
Event Agent Execution Speed	EAESU	

3.4 System Inputs

System inputs will be specified through a set of three Comma-Separated Values (CSV)[2] files which will be provided for each collection of clips[3]:

- The Event Database - *_EventDB.csv files
  This two-columns table defines the **EventID** and **EventName**.

- The Clip Database - *_ClipMD.csv files
  This five-columns table contains the metadata for each clip in the collection. The fields are **ClipID**, **MEDIA_FILE**, **CODEC**, **MD5SUM** and **DURATION**.

- The Trial Index Database - *_TrialIndex.csv files
  This three-columns table specifies the detection trials a system must perform. A detection trial is a (**ClipID, EventID**) pair for which a system must provide output. Each trial is identified with a unique **TrialID (composed as "ClipID.EventID")**. This file contains the full matrix of (ClipID, EventID) trials.

Two additional CSV tables are provided for training and system building:

- The Event Judgement Database - *_JudgementMD.csv files
  This multi-column file contains the clip-level event judgments (positive, near_miss, not_sure, and related) and additional annotations for (**ClipID, EventID**) pairs.

  This table does not contain the full matrix of pairs. Rather, it contains only the human judgments made during data collection.

- The Reference Database - *_Ref.csv files
  This two-column table specifies ground truth.

  *The file defines for each **TrialID** whether or not it is a "target" trial (contains an instance of the event) or a "non-target" trial (does not contain an instance of the event). This is represented in the **Targ** column as a 'y' or 'n' value. The file contains the full matrix of (**ClipID, EventID**) pairs using information derived from the Event Judgment Database.*

---

[2] See Appendix C for the CSV file format specification.
[3] These tables are the authoritative sources for system inputs. Participants should avoid using directory listings of the collections as inputs because the tables are an experimental control mechanism.

3.5 System Outputs and Documentation

For each submitted run to be considered complete and valid, participants must provide the information requested in this section. This requested information is a crucial element required for the research community to properly interpret the performance results.

3.5.1 System Description

The purpose of the system description document is to provide a list of the resources and techniques used to build the MED system and it identifies the computing resources and time required to process the test set.

Each submission must include a system description that describes:
- the assembled components,
- how closely the system addresses the <u>full MED goal of processing all test events,</u>
- the data used to train the event agent,
- the hardware components used and computation times of the metadata generation and event agent generation phases,
- the implementation method (automatic or semi-automatic) for the event agent generation, and the hardware components used, computational times, and the results of the event agent execution.

> See Appendix B section B.1 for the template that covers the *minimum* requirements of the system description document.

3.5.2 Event Agent Execution Reporting

A MED system processes the metadata store detecting instances of each event independently. For each system submission, two output files must be created using Experiment Identifiers (**EXPID**) which describe the characteristics of the run (see Appendix B for the definition of *EXPID*):

1. <EXPID>.threshold.csv

   Each line will contain information pertaining to the processing of a single event. Events not processed should not be included in the file. The 4 fields in this file are as follows.

   - **EventID**: The Event ID processed - copied from the event database file.
   - **DetectionThreshold**: A floating point probability value between 0 and 1 that is the detection score at or above which the system will assert that the event is detected in the clip. Values may differ across events. The threshold should be chosen to minimize errors based on an operating point with a 12.5:1 miss-to-false-alarm probability ratio. This ratio has been selected to support a typical search application where a user would not tolerate false alarms on the first page of output. The primary measure of performance will be based on clips at or above the threshold.
   - **DetectionTPT**: A value indicating the number of hours used during the *event agent execution phase* for the event.
   - **EAGTPT**: A value indicating the number of hours used during the *event agent generation phase* for the event.

2. <EXPID>.detection.csv

Each line will contain information pertaining to the processing of a single trial. The 2 fields in this file are as follows.

1. **TrialID**: The Trial ID processed - copied from the input trial index file.
2. **Score**: A probability value between 0 (low) and 1 (high) representing the system's confidence that the event is present in the clip. The value used should be used with the specifications of the **DetectionThreshold** defined earlier.

4   Data Resources

Internet multimedia data (i.e., video clips containing both audio and video) will be provided to registered MED participants. The data resources, collected by the Linguistic Data Consortium (LDC), consist of publicly available, user-generated content posted to Internet video hosting sites. The LDC will be the distribution point for the collection.

The video is provided in MPEG-4 formatted files. The video will be encoded to the H.264 standard. The audio will be encoded using MPEG-4's Advanced Audio Coding (AAC) standard.

See http://www.nist.gov/itl/iad/mig/med12.cfm for data licensing and acquisition instructions

MED data resources will consist of *training* resources which includes the event kits and "blind" *testing resources*.

4.1   Event Kits

There will be 25 events in MED12, 20 PS events and 5 AH events. The event names of the PS events are listed in Table 1.   The ten MED11 test events and ten new events comprise the PS events. The AH events will be revealed in the Fall according to the TRECVID Master Schedule.

Pre-Specified Events for MED 12	
Re-Tested MED11 Events	**New Events**
E006 - Birthday party	E021 – Attempting a bike trick
E007 - Changing a vehicle tire	E022 - Cleaning an appliance
E008 - Flash mob gathering	E023 - Dog show
E009 - Getting a vehicle unstuck	E024 - Giving directions to a location
E010 - Grooming an animal	E025 - Marriage proposal
E011 - Making a sandwich	E026 - Renovating a home
E012 - Parade	E027 - Rock climbing
E013 - Parkour	E028 - Town hall meeting
E014 - Repairing an appliance	E029 - Winning a race without a vehicle
E015 - Working on a sewing project	E030 - Working on a metal crafts project

Table 1: MED-12 Evaluation Events

Events will each be defined via an **event kit** consisting of:

event name	A mnemonic title for the event.
event definition	A textual definition of the event.

event explication	An expression of some event domain-specific knowledge needed by humans to understand the event definition.
evidential description	A textual listing of some attributes that are often indicative of an event instance. The evidential description provides a notion of some potential types of visual, acoustic, and temporal evidences indicating the event's existence. This is not an exhaustive list nor is it to be interpreted as required evidence.
illustrative video examples	Two forms of illustrative videos will be supplied: 1) "positive": (130) clips that contain at least 1 instance of the event 2) "related": (70) clips that contain one or more of the same or similar types of people, objects, locations, and/or actions associated with the target event, but do not meet the requirements to be a positive instance.

Table 2: Components of an event kit

Participants may use all resources supplied in the events kits, and any ancillary information provided in accompanying CVS tables, for system development and testing.

4.2 Training Resources

Training resources consist of three components: video collections, event kit video collections, and documentation. Participants may use the training resources for research, development testing, and error analysis of development testing. There are no MED12 evaluation rules or restrictions governing how participants are to use the training resources for MED system development.

4.2.1 Video Collections:

Three video collections are available for use:

- MED10 Pilot Training and Evaluation Collections
  The MED10 training and evaluation data collection contains annotations for three events: "assembling_a_shelter", "batting_in_a_run", and "making_a_cake". The data set consists of 94, 102, and 94 positive examples of clips respectively and 3178 background clips (near_miss examples are annotated in an accompanying metadata table). The video data can be found in LDC release LDC2011E06 as a combined collection (i.e., not divided by training and evaluation subsets).
- MED11 Development Collection
  The second training collection is the MED11 transparent development data collection (DEV-T). DEV-T is a ~350hr collection consisting of about 10K clips. The video data can be found in LDC released LDC2011E06 and LDC2011E41.
- MED11 Test Collection
  The third training collection is the MED11 Test Collection (MED11TEST). MED11TEST is a ~1000hr collection consisting of about 34K clips. The video data can be found in LDC release LDC2011E41.

4.2.2 Event Kit Collections

The event kit collections are available in three LDC releases:

- MED11 Events (E001-E015): LDC Release LDC2011E06 and LDC2011E92.
- MED12 Pre-Specified event kits (E021-E030): LDC Release LDC2012E01.
- MED12 Ad Hocevent kits (E016-E020): LDC Release LDC2012***TBD***.

4.2.3 MED Training Resource Documentation and Metadata

The LDC releases cumulative updates to the documentation and metadata for the MED development resources. The initial release for MED12 will be the LDC2012E02 release. It contains all textual data (event kit text files, databases, and annotations) for the development resources.

4.3 Testing Resources

The MED12 test collection is called the "Progress" Test (*PROGTEST*) collection. It will be used for *blind testing* for MED12, MED13, MED14, and MED15. PROGTEST is expected to be a 4000hr collection consisting of about 120K clips.

While PROGTEST will be used for *blind testing*, participants will have the data several months before the evaluation to build their metadata stores for MED12 and will retain the collection for the next three years. Therefore, participants must adhere to the following rules when interacting with the PROGTEST data:

- PROGTEST must be automatically processed to create the metadata store. Adaptation in the feature extraction process is permitted so long as the adaptation is fully automatic (**no human interaction**).
- Participants must not attempt to gain knowledge of PROGTEST properties or content by manually inspecting the video, clip metadata, output of the processing, or statistics developed during the processing[4].
- The PROGTEST metadata store must be frozen, (i.e., it may not be regenerated or augmented), before the first Pre-Specified event search occurs. It must remain frozen until the MED13 evaluation cycle begins.
- PROGTEST may only be searched against as part of an official evaluation submissions to NIST.
- PROGTEST video must be deleted from systems after the evaluation ends (November 1, 2012) and re-copied the following year.
- PROGTEST may not be used to influence the event agent generation phase.

5 Evaluation Measures

System output will be evaluated by how well the system detects MED events in PROGTEST and by the computing resources used to do so. The determination of correct detection will be at the clip level, i.e. systems will provide a response for each clip in PROGTEST. Each event will be scored independently.

MED system performance will be evaluated as a binary classification system by measuring performance of two error types: Missed Detection (MD) errors and False Alarm (FA) errors. NIST will report the primary performance measures for accuracy and processing speed, and a suite of diagnostic measures that may provide a deeper analysis of system performance.

5.1 Primary Measures

P_{MD} and P_{FA} for the event based on the *Detection Threshold*

   NIST will report the following performance statistics by applying the event-specific *Detection*

---

[4] An appropriate control method would be to have a non-team member process the MED11TEST data in preparation for the evaluation.

Threshold to the *Score* for each clip.

$P_{MD}(E, DT) = \#MD(E, DT) / \#Targets(E)$
$P_{FA}(E, DT) = \#FA(E, DT) / (\#TotalClips - \#Targets(E))$

Where

E	→ The event
DT	→ The detection threshold applied to the system's scores
#MD(E, DT)	→ The number of positive clips for event E < the Detection Threshold
#Targets(E)	→ The number of positive clips for event E.
#FA(E, DT)	→ The number of non-positive clips for event E >= the Detection Threshold
#TotalClips	→ The total number of clips in the testing collection

<u>Metadata Generation Processing Speed</u>

NIST will report both the **real-time** factor and the **single core adjusted real-time** factor to complete all steps necessary to build the metadata store.

Real-time factor is the Total Processing Time (TPT) for the process (as reported in the system description) divided by the number of hours of video in the test collection. TPT is the wall clock time (in hours) used during a computation phase, including I/O, from start to finish. The processing time for parallelized sub-steps adds to TPT as a single step. The processing time for metadata "shared" across sites, e.g., speech transcription, person tracking, etc. (including time used to incorporated data into the metadata store) adds to TPT as a single-step.

Single core adjusted real-time factor is an 'estimate' of the process' real-time factor for a single CPU core. It is computed by multiplying the TPT of each processing step by the number of CPU cores (counting hyper-threaded CPUs as a separate core) used and then divided by the number of hours in the test collection. The vagaries of queuing systems, threading, and CPU-internal parallel processing make this quantity onerous to compute accurately. Instead, the measure uses the course-grained "number of cores" as the adjustment factor for TPTs.

<u>Event Agent Generation Processing Speed</u>

NIST will report both the *real-time* factor and single *core adjusted real-time* factor for each event processed during the event agent generation phase.

<u>Event Agent Execution Processing Speed</u>

NIST will report both the *real-time* factor and single *core adjusted real-time* factor for each event processed during the event agent execution phase.

5.2 Diagnostic Measures

<u>Detection Error Tradeoff Curves</u>

Graphical performance assessment uses a Detection Error Tradeoff (DET) [2] curve that plots the system's Missed Detection probabilities (*P_{MD}*) and False Alarm probabilities (*P_{FA}*) over the full range of the system's decision scores. The resulting graph provides error performance characteristics for alternative operating points.

<u>Event-Weighted P_{MD} and P_{FA} based on the Detection Threshold</u>
NIST will report the P_{MD} and P_{FA} averaged over events.

<u>Confidence Score Quality Measures</u>
NIST will develop a confidence quality measure in collaboration with the community. This measure is yet to be determined.

<u>Area Under the Curve</u>
NIST will report the Area under the Receiver Operator Characteristic (ROC) curve. Note this is equivalent to the area **above** a DET curve.

<u>Relative Detection Threshold Error</u>
NIST will report the Relative Detection Threshold Error (RDTE) which measures the distance of the Detection Threshold (DetectThresh) from the Optimum Threshold (OptThresh) based on the 12.5:1 miss-to-false-alarm probability ratio line on the DET Curve.

RDTE = (OptThresh − DetectThresh) / OptThresh

5.3   Evaluation Tools and Command Line Example

NIST will use the Detection EVAluation (DEVA) tools within the NIST Framework for Detection Evaluation (F4DE) toolkit to score the evaluation submissions. The TRECVID MED12 Scoring Primer document (DEVA/doc/TRECVid-MED12-ScoringPrimer.html within the F4DE release) contains instructions for how to use the scorer. NIST will use the following command line to evaluate MED systems.

```
% DEVA_cli\
--profile MED12\
--outdir <OUTPUTDIR> \
--refcsv <DATA>_Ref.csv \
--syscsv <EXPID>.detection.csv:detection \
--syscsv <EXPID>.threshold.csv:threshold \
<DATA>_TrialIndex.csv:TrialIndex \
<DATA>_ClipMD.csv:ClipMD \
<DATA>_JudgementMD.csv:JudgementMD \
<DATA>_EventDB.csv:EventDB
```

Where,
- <OUTPUTDIR> is the pre-existing output directory
- <EXPID> is the Experimental ID defined in Appendix A
- <DATA> is the data set as defined in Appendix A

6   Result Submission Instructions

Submissions will be made via ftp according to the instructions in Appendix A. In addition to the system output, a system description must be supplied for each submission.

Participants may submit up to four "runs"[5] of system results and an additional two runs if the site participates in the Ad Hoc Pilot task. One, and only one, run for the Pre-Specified task must be designated as the primary submission and the rest as contrastive runs. Similarly, Sites participating in the Ad Hoc pilot must submit one primary Ad Hoc run and the rest of the Ad Hoc runs as contrastive runs. A primary run is expected to yield the best performance on the blind test set based solely on experiments using the training resources. NIST will focus the cross-site analysis on the primary runs.

7   Schedule

For TRECVID related schedule information please consult the main schedule on the TREVID 2012 web site.

8   References

[1] Harold W. Kuhn, "The Hungarian Method for the assignment problem", *Naval Research Logistic Quarterly*, **2**:83-97, 1955.

[2] Martin, A., Doddington, G., Kamm, T., Ordowski, M., Przybocki, M., "The DET Curve in Assessment of Detection Task Performance", Eurospeech 1997, pp 1895-1898.

---

[5] A run in this context is a particular configuration of technology components for an ensemble of MED events. For example, the inclusion or exclusion of acoustic features would be labeled as different runs but event-dependent parameter tuning would not. Participants should use their judgment as to whether or not technology configurations warrant differentiation.

Appendix A: Submission Instructions

The packaging and file naming conventions for MED12 relies on **Experiment Identifiers** (EXPID) to organize and identify the files for each evaluation condition and link the system inputs to system outputs. Since EXPIDs may be used in multiple contexts, some fields contain default values. The following section describes the EXPIDs.

The following EBNF describes the EXPID structure:

    EXPID ::= <TEAM>_MED12_<DATA>_<TASK>_<MEDTYPE>_<TRAINTYPE>_<EAG>_
        <SYSID>_<VERSION>
    where,

        <TEAM> ::= your Short TRECVID Team Name. Only alphanumerical characters are allowed, with no space.
        <DATA> ::= one of "MED12DRYRUN", "MED12TEST"
        <TASK> ::== either "PS" or "AH" for Pre-Specified or Ad-Hoc Task respectively as defined in Section 3.
        <MEDTYPE> :== either "MEDFull" or "MEDPart" Events Processed as defined in Section 3.
        <TRAINTYPE> :== either "EKFull" or "EK10Ex" Event Training conditions as defined in Section 3.
        <EAG> :== either "AutoEAG" or "SemiAutoEAG" specifying the Event Agent Processing style as defined in Section 3.
        <SYSID> ::= a site-specified string (that does not contain underscores) designating the system used.

        The SYSID string must be present. It is to begin with "p-" for the one and only primary system (i.e., your single best system) or with "c-" for any contrastive systems. It is then followed by an identifier for the system (only alphanumerical characters allowed, no spaces). For example, this string could be "p-baseline" or "c-contrast". This field is intended to differentiate between runs for the same evaluation condition. Therefore, a different SYSID should be used for runs where any changes were made to a system.

        <VERSION> ::= 1..n (with values greater than 1 indicating multiple runs of the same experiment/system)

Note: there can be only one primary system submission per (TEAM, DATA, TASK) combination.

In order to facilitate transmission to NIST and subsequent scoring, submissions must be made using the following protocol, consisting of four steps: (1) preparing a system description, (2) packaging system outputs and system descriptions, (3) running the submission checker to confirm validity of submission and (4) transmitting the data to NIST.

B.1 System Descriptions

Documenting each system is vital to interpreting evaluation results. As such, each submitted system, (determined by unique experiment identifiers), must be accompanied by a system description with the following information.

<div style="float:right; border:1px solid red;">
Jonathan Fiscus 5/25/12 1:05 PM
Deleted: .<EAGSPEED>_<MGSPEED>_<
</div>

<div style="float:right; border:1px solid red;">
Jonathan Fiscus 5/25/12 1:05 PM
Deleted: <EAESPEED> ::== either "EAESU" or "EAES1x" Event Agent Execute Speeds as specified in 3. .   ... [2]
</div>

Section 1   Experiment Identifier(s)
List all the EXPIDs for which system outputs were submitted. EXPIDs are described in further detail above.

Section 2   System Description
A brief technical description of your system; if a contrastive test, contrast with the primary system description. Please include contact information (name and email) for the submission.

Section 3   Metadata Generation System Hardware Description and Runtime Computation
Describe the hardware setup(s) to perform the metadata generation phase and report the Total Processing Time (TPT) and number of CPU cores as defined in Section 4.1. The phase may be broken down into sub-steps in which case the hardware, processing time, and CPU core counts of each sub-step must be documented.

A hardware setup is the aggregate of all computational components used to perform this phase. Examples of a system might be: a 16-node, Dual Quad Core 2.26 GHz Intel Xeon, 24GB RAM per node, with a 10TB Data Server.

Section 4   Event Agent Generation Hardware Description and Runtime Computation
Describes the computing hardware, including the number of CPU cores, used to generate the event agents(s). The Total Processing Time for each event will be placed in the system's *.threshold.csv file.

The hardware setup is the aggregate of all computational components used to perform this phase.

Section 5   Event Agent Execution Hardware Description
Describes the computing hardware, including the number of CPU cores, used for executing the event agent(s). The Total Processing Time for each event will be placed in the system's *.threshold.csv file.

The hardware setup is the aggregate of all computational components used to perform this phase. This hardware platform must be limited to a COTS standard personal computing platform.

Section 6   Training data and knowledge sources
Lists the resources used for system training, development, and runtime knowledge sources beyond the provided MED corpora.

Section 7   References
A list of pertinent references.

B.2 Packaging Submissions
All system output submissions must be formatted according to the following directory structure:
      output/<EXPID>/<EXPID>.txt
      output/<EXPID>/<EXPID>.detection.csv
      output/<EXPID>/<EXPID>.threshold.csv

      where,
            EXPID is the experiment identifier as described in Section B.1,
            <EXPID>.txt is the system description file as specified above (Section B.2),

<EXPID>.detection.csv is the CSV-formatted system output file containing the detection scores for each TrialID (see Section 2.2.2).
<EXPID>.threshold.csv is the CSV-formatted system output file containing the detection thresholds and processing speed measurements (see Section 2.2.2)

Multiple EXPIDs may be present under the "output/" directory.

B.3 Validating the Submission

The F4DE distribution contains a submission checker that validates the submission both at a syntactic and semantic level. Participants should check their submission prior to sending it to NIST. NIST will reject submissions that do not pass validation. The TRECVID MED12 Scoring Primer document (DEVA/doc/TRECVid-MED12-ScoringPrimer.html within the F4DE release) contains instructions for how to use the validator. NIST will use a similar command line to validate MED submission files:

```
%TV12MED-SubmissionChecker --TrialIndex <DATA>_TrialIndex.csv \
    MED12_<TEAM>_<DATA>_<SUBNUM>.tar.bz2
```

B.4 Transmitting Submissions

To prepare your submission, first create the previously described file/directory structure. The following instructions assume that you are using the UNIX operating system. If you do not have access to UNIX utilities or ftp, please contact NIST to make alternate arrangements.

First, change directory to the parent directory of your "output/" directory. Next, type the following command:

```
tar -cvf - ./output | gzip > MED12_<TEAM>_<DATA>_<SUBNUM>.tgz
where,
        <TEAM> and <DATA> and are the same as defined above.
        <SUBNUM> is an integer 1 to n, where 1 identifies your first submission, 2 your second,
        etc.
```

Important: only the latest submission will be used for scoring, but a submission file can contain multiple EXPIDs as long as there is only one primary one. If there is only one EXPID, it must be the primary one.

This command creates a single tar/gzip file containing all of your results. After shipment to NIST (as described in the next step), NIST will validate your submission with a syntactic and semantic validator.

Next, ftp to jaguar.ncsl.nist.gov giving the username 'anonymous' and (if requested) your e-mail address as the password. After you are logged in, issue the following set of commands, (the prompt will be 'ftp>'):

```
ftp> cd incoming
ftp> binary
ftp> epsv4 off
ftp> passive auto
ftp> put MED12_<TEAM>_<DATA>_ <SUBNUM>.tgz
ftp> quit
```

Note: the "epsv4 off' is designed to bypass a limitation of the FTP server, and should return: "EPSV/EPRT on IPv4 off.". If your prompt returns something different it is likely that your system does not support this feature, and therefore it is not needed.

Note: the "passive auto" should return "Passive mode: on; fallback to active mode: on".

Note that because the "incoming" ftp directory (where you just ftp-ed your submission) is write protected, you will not be able to overwrite any existing file by the same name (you will get an error message if you try), and you will not be able to list the incoming directory (i.e., with the "ls" or "dir" commands). Please note whether you get any error messages from the ftp process when you execute the ftp commands stated above and report them to NIST.

The last thing you need to do is send an e-mail message to med_poc@nist.gov to notify NIST of your submission. The following information should be included in your email:
- the name of your submission file,
- the file size,
- a listing of each of your submitted experiment IDs.

Please submit your files in time for us to deal with any transmission errors that might occur well before the due date if possible. Note that submissions received after the stated due dates for any reason will be marked late.

Appendix C: Comma Separated Value File Format Specifications

The MED evaluation infrastructure uses Comma Separated Value (CSV) formatted files with an initial field header line as the data interchange format for all textual data. The EBNF structure the infrastructure uses is as follows:

CSVFILE :== <HEADER> <DATA>*

<HEADER>	:==	<VALUE> {"," <VALUE> }* <NEWLINE>
<DATA>	:==	<VALUE> {"," <VALUE> }* <NEWLINE>
<VALUE>	:==	<DOUBLEQUOTE><TEXT_STRING><DOUBLEQUOTE>

The first data record in the files is a header line. The header lines are required by the evaluation infrastructure.

Each header and data record in the table is one line of the text file. Each field value is delimited by double quotes and is separated from the next value with a comma.

An example trial index is (*_TrialIndex.csv):

```
"TrialID","ClipID","EventID"
"72.P001","72","P001"
"72.P002","72","P002"
"72.P003","72","P003"
"285.P001","285","P001"
"285.P002","285","P002"
"285.P003","285","P003"
```

An example system output file is (*.detection.csv):

```
"TrialID","Score"
"72.P001","0.062712"
"72.P002","0.978791"
"72.P003","0.115392"
"285.P001","0.801007"
"285.P002","0.861036"
"285.P003","0.120700"
```

An example threshold system output file is (*.threshold.csv):

```
"EventID","DectectionThreshold","DetectionTPT","EAGTPT"
"P001","0.54","5923.3","12.5623"
"P002","0.74","9204.3","25.4534"
"P003","0.66","3456.2","19.7683"
```