

executable *parameter_file image_directory target_sigset query_sigset similarity_file
quality_target_sigset quality_query_sigset*

Executable Arguments

Table A 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. For the 1:1 and 1:many matching experiments, this will be the primary output data structure. This will be an input data structure for the similarity normalization experiment.
quality_target_sigset	Output (optional)	Sigset	The name of the target sigset to which quality scores should be written. This file should be a copy of the input target sigset with quality scores added. Ignore this parameter if your executable does not compute quality scores.
quality_query_sigset	Output (optional)	Sigset	The name of the query sigset to which quality scores should be written. This file should be a copy of the input query sigset with quality scores added. Ignore this parameter if your executable does not compute quality scores.

Table A Description of executable arguments

Directory Structure

Each participant will be given the directory structure shown in Figure A. All experiments (and thus executables) will be instantiated from the top level (*/algorithm_name*) directory.

Executables must be installed to the */bin* directory. Any libraries needed by executables should be installed in the */lib* directory. After installation, both the */bin* and */lib* directory will be made read and execute only. All required outputs (similarity matrices, signature sets, etc) generated during experiment execution should be written to the */output* directory. Users will be given write permission to the */output* directory. The */temp* directory should be used for any temporary files generated during program execution. Users will be given read and write privileges to the */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 in Table A. Users will be given the relative path to the images via the *image_directory* parameter.

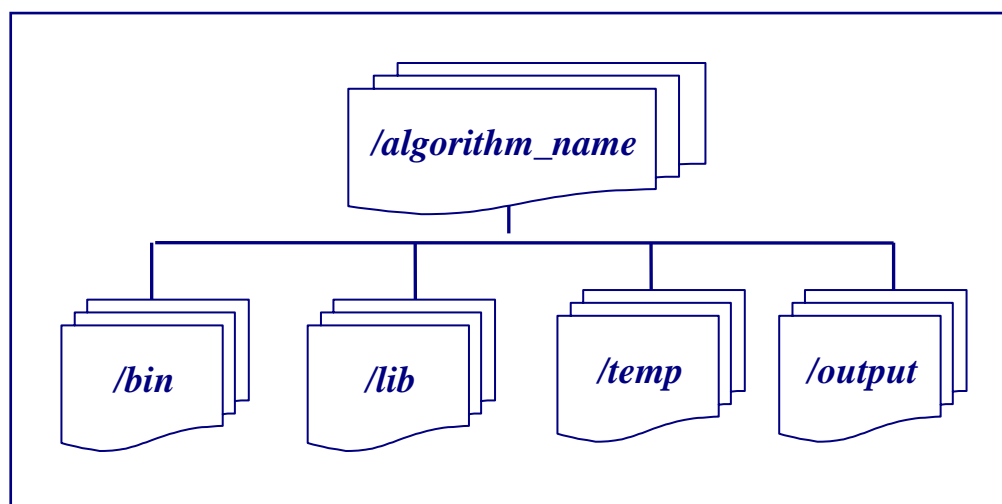


Figure A ICE 2006 directory structure.

Important Notes

- The name of the */algorithm_name* directory will correspond to the name of your submitted executables which must adhere to the ICE 2006 algorithm naming convention.
- Algorithms will be instantiated (run) from the */algorithm_name* directory.
- Executables should reside in the */bin* directory.
- Do not hard code any drive information. Specifically, do not assume that the */algorithm_name* directory will be on the C: drive.
- The output directory will be directly below the */algorithm_name* directory. Do not create a */output/output* or */bin/output* directory.

- Any modifications to the \$path or the \$ld_library_path variables should be done by your install script and should be temporary (i.e. should only effect the current command or terminal console).

Image Directory

The image directory is a string that specifies the relative path to 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 a image named ‘*tic\image1.tiff*’, you should process the image in the file ‘..\images*tic\image1.tiff*’. Note: 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.

Parameter Files

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.

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 B. 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 three attributes named *name*, *type* and *feature_extraction_mode* respectively. For the ICE experiments the *type* attribute will have a value of “1-1” and the value of the *feature_extraction_mode* attribute will be “Full”. The *Experiment* element has three child elements *Target*, *Query* and *LogFile*. Both the *Target* and *Query* elements will have one attribute: *capture_device*. The value “LG2200” will refer to images similar to those captured at the University of Notre Dame. Attribute values and information describing other capture devices may be distributed at a later time. The *LogFile* element specifies where your algorithm should write status and debug information.

```

<?xml version="1.0"?>
<Experiment name="1_to_1" type="1-1" feature_extraction_mode="Full" >
  <Target capture_device="LG2200" />
  <Query capture_device="LG2200" />
  <LogFile name="1_to_1_logfile.txt" />
</Experiment>

```

Figure B Example of a parameter file

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	The type of task and executable that this is experiment is being evaluating.
feature_extraction_mode	enumerated	Full	The enumerated values of this attribute specify whether ground truth data is provided to aid in feature extraction. For ICE 2006 the value will be "Full" which indicates that ground truth data is not provided.
capture_device	enumerated	LG2200 TBD	The enumerated values of this attribute specify the type of sensor used for image capture.
Name (LogFile)	string	Filename	All logging information generated by the program should be written to this file in the output directory.

Table B Description of elements in the similarity header.

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 ICE 2006 participants. Parsers will not be made available for other languages (e.g. Matlab). However, users should be able to easily create parsers in other languages using the C++ classes as a guide.

Writing

ICE users will not be required to write (output) parameter files.

Signature Sets (Sigsets)

Overview

Signature Sets (Sigsets) are the primary input structure for ICE 2006. They will use used to list the files in the Target and/or Query sets.

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 two *Presentation* child elements that correspond to the iris image(s).

Figure C illustrates the simple Signature Set structure with elements depicted blue, attributes depicted red (lowercase) and attribute values depicted black (uppercase and in quotes). This example has four **biometric-signature** elements that represent three subjects. (There are two presentations for subject “557”.) The signatures are named “557”, “558” and “559”. Each signature has one **presentation** (image) of a left iris. All images are tiff files.

```
<?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">

  <biometric-signature name="557">
    <presentation name="242311" modality="IRIS-LEFT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="557">
    <presentation name="242979" modality="IRIS-LEFT" file-name="lg2200/242979.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="558">
    <presentation name="243902" modality="IRIS-LEFT" file-name="lg2200/243902.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="559">
    <presentation name="241364" modality="IRIS-LEFT" file-name="lg2200/241364.tif" file-format="tiff"/>
  </biometric-signature>
</biometric-signature-set>
```

Figure C Example of a Signature Set with one left iris image per subject.

Figure D illustrates the general Signature Set structure with elements depicted blue, attributes depicted red (lowercase) and attribute values depicted black (uppercase and in quotes). This example has four **complex-biometric-signature** elements that represent three subjects. (Note: While **presentation** elements grouped under a single **complex-biometric-signature** are guaranteed to correspond to the same subject, separate signatures may correspond to the same subject.) The signatures are named “557”, “558” and “559”. Signature “557” has two **presentation** (capture sessions) labeled “242311” and “242979” respectively. Each signature consists of left iris image and a right iris images which are specified with the modality attribute in the presentation element. All images are stored as tiff images.

```
<?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="557">
    <presentation name="242311" modality="IRIS-LEFT" file-name="lg2200/242311.tif" file-format="tiff"/>
    <presentation name="242321" modality="IRIS-RIGHT" file-name="lg2200/242321.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="557">
    <presentation name="242979" modality="IRIS-LEFT" file-name="lg2200/242979.tif" file-format="tiff"/>
    <presentation name="242334" modality="IRIS-RIGHT" file-name="lg2200/242334.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="558">
    <presentation name="243902" modality="IRIS-LEFT" file-name="lg2200/243902.tif" file-format="tiff"/>
    <presentation name="242421" modality="IRIS-RIGHT" file-name="lg2200/242421.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="559">
    <presentation name="241364" modality="IRIS-LEFT" file-name="lg2200/241364.tif" file-format="tiff"/>
    <presentation name="a42319" modality="IRIS-RIGHT" file-name="lg2200/a42319.tif" file-format="tiff"/>
  </complex-biometric-signature>
</biometric-signature-set>
```

Figure D Example of a Signature Set with two iris images per subject.

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 ICE 2006 participants. We also provide examples of 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 when using this simplified version of the Similarity structure. Parsers are not provided for other languages (e.g. Matlab). However, users should be able to create a Java wrapper that parses the Signature Set and passes the appropriate data structures to these executables.

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.

Signature Sets (Sigsets) with Quality Scores

Overview

All Signature Sets (Sigsets) has an optional element at the signature level that is used to specify a quality score for the signature. The value of the score will be a positive integer where larger values correspond to better images. This will be the output format created by executables.

Structure

The structure of Signature Sets with quality scores is identical to the standard Signature Set format with the exception of the added quality score. The quality score is an attribute of the biometric-signature and complex-biometric-signature elements. It is important to note that quality scores correspond to signatures. Thus, in the case of complex-biometric-signature elements the score should be representative of both presentations in the signature. Figure E and Figure F depict example output Signature Sets with quality for the input Signature Sets depicted previously in Figure C and Figure D.

```

<?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">

  <biometric-signature name="557" quality="98" >
    <presentation name="242311" modality="IRIS-LEFT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="557" quality="90" >
    <presentation name="242979" modality="IRIS-LEFT" file-name="lg2200/242979.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="558" quality="100" >
    <presentation name="243902" modality="IRIS-LEFT" file-name="lg2200/243902.tif" file-format="tiff"/>
  </biometric-signature>
  <biometric-signature name="559" quality="80" >
    <presentation name="241364" modality="IRIS-LEFT" file-name="lg2200/241364.tif" file-format="tiff"/>
  </biometric-signature>
</biometric-signature-set>

```

Figure E Example of a Signature Set with one left iris image per subject.

```

<?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="557" quality="72" >
    <presentation name="242311" modality="IRIS-LEFT" file-name="lg2200/242311.tif" file-format="tiff"/>
    <presentation name="242311" modality="IRIS-RIGHT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="557" quality="88" >
    <presentation name="242979" modality="IRIS-LEFT" file-name="lg2200/242979.tif" file-format="tiff"/>
    <presentation name="242311" modality="IRIS-RIGHT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="558" quality="94" >
    <presentation name="243902" modality="IRIS-LEFT" file-name="lg2200/243902.tif" file-format="tiff"/>
    <presentation name="242311" modality="IRIS-RIGHT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </complex-biometric-signature>
  <complex-biometric-signature name="559" quality="98" >
    <presentation name="241364" modality="IRIS-LEFT" file-name="lg2200/241364.tif" file-format="tiff"/>
    <presentation name="242311" modality="IRIS-RIGHT" file-name="lg2200/242311.tif" file-format="tiff"/>
  </complex-biometric-signature>
</biometric-signature-set>

```

Figure F Example of a Signature Set with two iris images per subject.

Parsing

While the standard tools for parsing Signature Sets can be used to parse Signature Sets with quality, the ICE 2006 protocol does not require users to parse Signature Sets with quality.

Writing

Like most XML documents, Signature Sets with quality 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.

Similarity Matrices

Overview

Similarity Matrices are the primary output structures of recognition algorithms in ICE 2006. 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.

Structure

The Similarity Matrix is similar to many image files in that it contains a textual header prepended to a binary representation of a $N \times M$ data structure. The structure of the header is depicted below in Figure G. 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 an end-of-line character. Table C describes the elements in the similarity header.

```

D2
BEE_DIST/signature_sets/ICE_Exp_1.0.1_Target.xml
BEE_DIST/signature_sets/ICE_Exp_1.0.1_Query.xml
MF 1425 1425 xV4

```

Figure G 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'	eol	The value '2' corresponds to the version of similarity matrix.
Target name	string	eol	This string should be the same as the name of the target sigset provided to the matching algorithm.
Query name	string	eol	This string should be the same as the name of the query sigset provided to the matching algorithm.
Format	The characters 'MF'	space	The values correspond to a matrix, 'M', containing float, 'F', values.
Rows	integer	space	The number of signatures in the query set.
Cols	integer	space	The number of signatures in the target set.
Magic number	0x12345678 (4-bytes binary)	eol	This binary value is used to check for Endian (byte swapping).

Table C 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 image to each of the target images. There must not be any white space characters separating scores in the body of the Similarity Matrix.

Important Notes

- The dimensions of the similarity matrix corresponds to the number of signature elements in the target and query sigsets not the number of presentations.

Parsing

C++ and Java classes for parsing Similarity Matrices are provided in the BEE (Biometric Experimentation Environment) distribution. These classes will be available to ICE 2006 participants. Parsers will not be made available for other languages (e.g. Matlab). However, users should be able to easily create parsers in other language using the C++ classes as guides.

Writing

C++ and Java classes for writing Similarity Matrices are provided in the BEE (Biometric Experimentation Environment) distribution. These classes are available to ICE 2006 participants. Parsers will not be made available for other languages (e.g. Matlab). However, users should be able to easily write Similarity Matrices in other language using the C++ classes as guides.

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 we (with your limited support) can easily and quickly troubleshoot your algorithm. Since we will have limited time to troubleshoot programs, please provide detailed debug messages in your log file.

Example of Calling Signature for ICE

```
executable_name /home/ice/parameters/parameter_exp1.xml /home/ice/images/  
/home/ice/sigsets/target_exp1.xml /home/ice/sigsets/query_exp1.xml  
output/similarity_exp1.mtx xml output/quality_target_exp1.xml  
output/quality_query_exp1.xml
```

Clearly, this entire command would be on one line. Notice that both absolute and relative directories can be used.