BEL Framework V1.2 Tools User Guide

Table of Contents

Introduction

This user guide will help you understand the use and concepts underlying Selventa's BEL Framework Tools. The BEL Framework Tools are a collection of applications that are distributed along with the BEL Framework and can be used to manage various aspects of working with the BEL Framework.

Other components of the BEL Framework include:

- BEL Workbench

- BEL Framework Compiler/Assembler

- BEL Framework Java API

- BEL Framework Web API

These components are documented separately.

Version Changes

There are no version changes associated with this document.

System Requirements

The BEL Framework Tools are a component of the BEL Framework and share the same system requirements. Please refer to the *BEL Framework V1.2 Configuration Guide* for a complete list of system requirements.

Tools Overview

The BEL Framework Tools are a suite of utility applications that can help you manage a BEL Framework installation. This version of the BEL Framework is distributed with the following eight (8) tools.

Document Management Tools
- BEL Document Validator

- DocStore Summarizer

- DocStore Importer

- DocStore Exporter

Kam Tools
- Kam Summarizer

- Kam Exporter

- Kam Importer

Other Tools
- Cache Manager

The use of each of these tools is described in the following sections.

Document Management Tools

The BEL Framework Document Store (DocStore) is a database that contains BEL Documents that have been created using the BEL Workbench application or imported for use with the BEL Workbench.

The Document Management Tools provide utilities for validating, importing, exporting, and summarizing BEL Documents added to the DocStore.

Note: The import and export utilities are command line versions of the import and export functions available through the BEL Workbench.

BEL Document Validator

The BEL Document Validator tool, BelCheck, performs syntax and semantic validation on a BEL Document.

Usage

```
BelCheck.sh [FILE}
```

where:

[FILE] is the name of the BEL Script or XBEL Documents to validate.

Command Line Arguments

This section identifies the command line arguments for the BEL Document Validator.

Argument	Description
-h --help	Shows command line options.
--permissive	Enables permissive mode. This mode will return a success code if syntactic or semantic warnings are detected in the input file. The default is to disable permissive mode.
--pedantic	Enables pedantic mode. This mode treats warnings as errors and will return a failure code if a syntax or semantic warning is detected in the input file. The default is to disable pedantic mode.
-q --quiet	Enables quiet mode. This mode suppresses warnings and errors sent to the output. The default is to disable quite mode.
--summary	Optional. This switch will output the number of warnings and errors detected in the input file. The summary will still be written if the --quiet switch is set.

Examples

<u>Validating a BEL Script document</u>

```
./BelCheck test.bel --permissive -q
```

This command will read the BEL Script document `test.bel` and check it for syntactic and semantic errors. The `--permissive` switch will force the validator to return a success code

if one or more warnings are generated during the validation process. The `-q` switch will suppress listing the warnings on `stdout`.

DocStore Summarizer

The DocStore Summarizer tool provides a summary of BEL Documents stored in the DocStore.

Usage

```
DocstoreSummarizer.sh
```

Command Line Arguments

This section identifies the command line arguments for the Docstore Summarizer.

Argument	Description
-h --help	Shows command line options.
-l --list	List the BEL Documents stored in the Document Store.
-n [name] --document-name	Name of a BEL Document to provide a detailed summary for.
-g [docgroup name] --document-group-name	Name of a Document Group in the BEL Document Store to provide a detailed summary for.
-o [file] --output-file	Optional. If present, it indicates the name of the output file to write the summary to. The file will be written as XHTML. The default is to write the summary to the console.
--no-preserve	If present, if a file with the same name as the output file already exists, it will be overwritten. The default is to preserve the existing file and to generate a warning.
-s [filename] --system-config-file	Optional. Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. The default is to use the default systems configuration file found in [installdir]/config.
--debug	Enables debug mode. The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs. The default is to disable verbose mode.

Examples

Listing BEL Documents in a Document Store

```
./DocstoreSummarizer.sh -l
```

Summarizing a BEL Document

```
./DocstoreSummarizer.sh -n "Test Document"
```

```
./DocstoreSummarizer.sh –g "Doc Group 1"
```

DocStore Importer

The Docstore Importer tool provides a command-line utility for adding BEL Documents to a BEL Document Store. This tool provides an alternative to the document import functions available through the BEL Workbench and can handle documents larger than 20 mb.

Usage

```
DocstoreImport.sh [-g <docgroup name>] [FILE ...]
```

where:

`[FILE …]` is a list of BEL Documents to import.

Command Line Arguments

This section identifies the command line arguments for the Docstore Summarizer.

Argument	Description
-h --help	Shows command line options.
-g [docgroup name] --document-group-name	Optional. The name of a Document Group in the Document Store to add the BEL Document(s) to. If not specified, the BEL Document(s) will be added without being assigned to any Document Groups. If the document group does not exist it will be created.
--no-preserve	If present, if a BEL Document with the same name and version exists in the Document Store it will be overwritten. The default is to preserve the existing BEL Document in the Document Store and generate a warning.
--permissive	Enables permissive mode. This mode will allow the import of documents that contain syntax errors at the BEL Statement level. The default is to disable permissive mode.
--pedantic	Enables pedantic mode. This mode treats warnings as errors and will terminate the import of a BEL Document if a syntax or semantic warning is detected in the input file. The default is to disable pedantic mode.
-s [filename] --system-config-file	Optional. Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. The default is to use the default systems configuration file found in [installdir]/config.
--debug	Enables debug mode. The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs.

The default is to disable verbose mode.

Examples

Importing an XBEL Document

```
./DocstoreImport.sh –permissive –g "Test Group"
~/BELDocuments/*.bel
```

This command will import all the BEL Script documents stored in the `~/BELDocuments` folder and add them to a Document group called `Test Group`. The Document Group will be created if it does not already exist. The `--permissive` switch will force the imported to continue to import any BEL Documents which have syntactic or semantic errors.

DocStore Exporter

The Docstore Export tool provides a command-line utility for exporting BEL Documents stored in a Document Store to XBEL or BEL Script files. This utility provides an alternative to the document export functions available through the BEL Workbench.

Usage

```
DocstoreExport.sh
```

Command Line Arguments

This section identifies the command line arguments for the Docstore Summarizer.

Argument	Description
-h --help	Shows command line options.
-o [output directory] --output-directory	The path to a directory where the BEL Document file(s) will be written. This directory must exist and be writable.
-n [Doc1, Doc2,…]	Optional. A list of BEL Documents to export.
-g [docgroup name] --document-group-name	Optional. The name of a Document Group in the BEL Document store to export the BEL Document(s) from.
-x --xbel	Write all BEL Documents using XBEL format. This is the default.
-b --bel-script	Write all BEL Documents using BEL Script format. The default is to write as XBEL format.
--no-preserve	If present, if a BEL Document with the same name exists in the output directory it will be overwritten. The default is to preserve the BEL Document in the output directory and to generate a warning.
-s [filename] --system-config-file	Optional. Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. The default is to use the default systems configuration file found in [installdir]/config.
--debug	Enables debug mode. The default is to disable debug mode.

-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs.
	The default is to disable verbose mode.

Examples

Exporting a Single BEL Document

```
./DocstoreExport.sh –n "Small Corpus" –b –o ~/BELDocuments
```

This command will export the BEL Document named `Small Corpus` in the DocStore in BEL Script format and put the file in the folder called `~/BELDocuments`. If a file with the same name exists in the folder, a warning will be generated.

Exporting a BEL Document Group

```
./DocstoreExport.sh –g "Test Group" –x –o ~/BELDocuments –no-
preserve
```

This command will export all the BEL Documents in the DocStore that are part of the `Test Group` Document group. The Documents will be written in XBEL format and placed in the folder called `~/BELDocuments`. Any files that already exist in the folder with the same name will be overwritten.

KAM Tools

The BEL Framework KAM Database (KamStore) is a database that is used by the BEL Framework to store KAMs that have been created.

The KAM Tools are utilities or summarizing KAMs, exporting KAMs in various formats and importing KAMs into a KamStore.

KAM Summarizer

The KAM Summarizer tool provides a summary of the KAMs stored in a KamStore. The KAM Summarizer outputs summaries in XHTML format so it can be quickly formatted for use in other systems.

Usage

```
KamSummarizer.sh
```

Command Line Arguments

This section identifies the command line arguments for the KAM Summarizer.

Argument	Description
-h --help	Shows command line options.
-l --list	List the KAMs stored in the KamStore.
-k [KAM] --kam-name	Name of a KAM to provide a detailed summary for.
-o [file] --output-file	Optional. If present, it indicates the name of the output file to write the summary to. The file will be written as XHTML.
	The default is to write the summary to the console.

-s [filename] --system-config-file	Optional. Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. The default is to use the default systems configuration file found in [installdir]/config.
--no-preserve	Optional. If present, if a file with the same name as the output file already exists, it will be overwritten. The default is to preserve the existing file and to generate a warning.
--debug	Enables debug mode. The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs. The default is to disable verbose mode.

Examples

<u>Listing KAMS in a KAM Store</u>

```
./KamSummarizer.sh –l –o kam_catalog.xhtml
```

This command will list the KAMs available in the KAM Store and write them to an XHTML file called `kam_catalog.xhtml`.

<u>Summarizing a KAM</u>

```
./KamSummarizer.sh –k test
```

This command will summarize the KAM named `test` and write the summary to the system console.

KAM Exporter

The KAM Exporter tool will export a KAM to a file for use with other systems. The current version supports a portable KAM format, XGMML, and N3 RDF format.

Usage

```
KamExporter.sh
```

Command Line Arguments

This section identifies the command line arguments for the KAM Exporter.

Argument	Description
-h --help	Shows command line options.
-l --list	List the KAMs stored in the KamStore.
-k [KAM] --kam-name	Name of a KAM to be exported.
-o [file] --output-file	File path for the output file that will be created.
-t [KAM, XGMML or RDF]	Specifies the type of export file to generate. If XGMML the

--type	KAM will be exported in XGMML format. If RDF the KAM will be exported as an RDF using N3 format. If KAM is specified, the KAM will be exported in a portable binary format that can be subsequently imported to another KAM Store using the KamImporter tool.
	The default is to export to a portable KAM format.
-p [password] --password	Optional. If present, the password is used to encrypt the portable KAM file. The same password must be used to import the portable KAM with the Kam Importer tool.
	The default is to export the KAM without encryption.
--no-preserve	Optional. If present, if a file with the same name as the output file already exists, it will be overwritten.
	The default is to preserve the existing file and to generate a warning.
-s [filename] --system-config-file	Optional. Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration.
	The default is to use the default systems configuration file found in [installdir]/config.
--debug	Enables debug mode.
	The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs.
	The default is to disable verbose mode.

Examples

Exporting a KAM to XGMML

```
./KamExporter.sh –k test –t XGMML –o test.xgmml –no-preserve
```

This command will export the KAM named `test` in the KAM Store in XGMML format to a file named `test.xgmml`. If the file already exists it will be overwritten.

Exporting a KAM to RDF

```
./KamExporter.sh –k test –t RDF –o test.n3
```

This command will export the KAM named `test` in the KAM Store in N3 RDF format to a file named `test.n3`. If the file already exists a warning will be provided and the command will exit.

Exporting a KAM to a Portable Format

```
./KamExporter.sh –k test –t KAM –p secret –o test.kam –no-
preserve
```

This command will export the KAM named `test` in the KAM Store in the portable KAM format to a file named `test.kam`. The KAM will be encrypted using the password `secret`. If a file of the same name already exists it will be overwritten.

KAM Importer

The KAM Importer utility will import a portable KAM to a KamStore. The KAM must have been previously exported to a portable binary format using the KAM Exporter utility.

Usage

```
KamImporter.sh [FILE]
```

where:

[FILE] is the path to a portable binary KAM.

Command Line Arguments

This section identifies the command line arguments for the KAM Importer.

Argument	Description
-h --help	Shows command line options.
-l --list	List the KAMs stored in the KamStore.
-k [KAM] --kam-name	Optional. If present the imported KAM will be renamed to this name in the KAMStore. The default is to use the existing name for the KAM being imported.
-p [password] --password	Optional. If present, the password is used to decrypt the portable KAM file if the file was encrypted during export.
--no-preserve	Optional. If present, if a KAM with the same name as the KAM being imported already exists in the KAMStore, it will be overwritten. The default is to preserve the existing KAM and to generate a warning.
-s [filename] --system-config-file	Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. The default is to use the default systems configuration file found in [installdir]/config.
--debug	Enables debug mode. The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs. The default is to disable verbose mode.

Examples

Importing a KAM

```
./KamImporter.sh -k "new kam"-p secret test.kam -no-preserve
```

This command will import the KAM in the portable KAM file named test.kam to the KAM Store using the password secret to decrypt the KAM. The KAM will be renamed to new kam. If a KAM with the name new kam already exists in the KAM Store it will be overwritten.

Other Tools

The tools folder in the BEL Framework distribution contains other tools that can help you manage the BEL Framework installation.

Cache Manager

The Cache Manager can be used to manage the cache of information that is created and used by the BEL Compiler and BEL Workbench applications.

The Cache Manager can perform the following actions:

- List the contents of the cache

- Remove all cached documents (purge)

- Replace all cached documents (refresh)

- Update any cached document

Usage

```
CacheManager [-l] [-c <resource file>] [-a <resource index file>]
[-u ] [-p] [-c <filename>]
```

Command Line Arguments

This section identifies the command line arguments for the Cache Manger.

Argument	Description
-h --help	Shows command line options.
-l --list	List the items stored in the cache.
-p --purge	Purges the contents of the cache. All resources in the cache will be removed.
-u --update	Updates all cached resources. The cache manager will attempt to download the latest version of a cached resource and update the cache with the new information.
-i [url] --index-file	Read the index file specified by the URL and cache all resources identified in the index file. If the URL is not specified the index file referenced using the resource_index_url of the system configuration file is used.
-n [url] --new-from-index-file	Read the index file specified by the URL and cache all resources identified in the index file that are not currently cached. If the URL is not specified the index file referenced using the resource_index_url of the system configuration file is used.
-a [url] --add	Add the resource specified by url to the cache. If the resource already exists in the cache it will be updated.
-g [file] --generate-checksum	Generate a checksum for the specified file.
-s [filename] --system-config-file	Allows the user to specify the path to a system configuration file to use to specify the BEL Framework configuration. Default is to use the default systems configuration file found

in [installdir]/config.

--debug	Enables debug mode.
	The default is to disable debug mode.
-v --verbose	Enables verbose mode. This mode outputs additional information as the application runs.
	The default is to disable verbose mode.

Additional Information

This section provides additional information that might be helpful to you.

Obtaining Technical Support

Technical support is available by phone or email during normal business hours (8am to 5pm EST).

Email Support

Send an email to support@selventa.com. Please make sure to include your name, a phone number where you can be reached, and details about the issue.

Phone Support

Please call Selventa's technical support line at (617) 851-5273 during normal support hours.

Learning More About Selventa's Software and Services

For all sales and other inquires, please contact:

Louis Latino
EVP Sales and Marketing
One Alewife Center,
Cambridge MA 02140

Phone: (617) 547-5421 x237
Email: llatino@selventa.com