

NCI LexEVS Data Packaging and Deployment Utility

Contents of this Page

- [NCI LexEVS Data Packaging and Deployment Utility](#)
- [Usage Information](#)
 - [Technical Overview](#)
 - [Instruction set for "package.sh"](#)
 - [Instruction set for "deploy.sh"](#)
- [Property files use to determine configurations for packaging and deployments.](#)
 - [package.sh: Properties available for package.properties](#)
 - [deploy.sh: Properties available for deploy.properties](#)
- [Downloads](#)
- [Install the NCI LexEVS Data Packaging and Deployment Utility](#)
- [Additional Notes](#)
 - [Stopping/starting Jboss](#)
 - [Checking if Jboss has started](#)
 - [Example packaging of terminology:](#)
 - [Example database and file system deployment.](#)

NCI LexEVS Data Packaging and Deployment Utility

The NCI LexEVS Data Packaging and Deployment Utility is primarily intended to ease the manual process of packing and publishing LexEVS 5.1 data from environment to environment (environment here refers to software development tiers such as Quality Assurance). In general, LexEVS 5.1 data is loaded in to a LexEVS 5.1 environment using the LexEVS 5.1 loader and using the LexEVS Data Packaging and Deployment utility it is possible to package the loaded data for deployment and to then to deploy said data to other LexEVS 5.1 environments. Updates of the associated "metadata.xml", "registry.xml" and Lucene index files are handled automatically by this utility. This utility will also handle the removal of selected terminologies from target environments.

Usage Information

The NCI LexEVS Data Packaging and Deployment Utility contains two separate executable files, the first one being "package.sh" which invokes a Java executable which is able to process user packaging commands (an instruction set) for the purpose of creating a "deployment" package. The user is able to select which terminologies are to be deployed and which terminologies are to be removed using the "package.sh" utility. The second, "deploy.sh" invokes a Java executable that is able to read the previously created package and to deploy it to a target environment.

Technical Overview

The NCI LexEVS Data Packaging and Deployment Utility is written as a Java command line application following the standard Unix command line syntax and base on NCI BDA-Lite. NCI BDA-Lite is a light weight application build and deployment library enabling developers to easily incorporate common software development tasks such as application server configuration and web application deployment via custom Apache Ant Task(s). In addition, the BDA-Lite Java API allows developers to execute deployment tasks outside of the Apache Ant environment via plain Java applications such as with the NCI LexEVS Data Packaging and Deployment Utility.

Below is a listing of the instruction set for both the packaging and deployment utilities:

Instruction set for "package.sh"

- (H)elp - Display a listing of all available commands.
- (L)ist - Displays a numbered listing of available terminologies (for packaging) from the source environment. The source environment is determined by the value of property "master_lbconfig" in file "package.properties". Please see [EVS:package.properties](#) for available packaging properties.
- (S)elect # - Selects terminology '#' to be packaged.
- (U)nselect # - Unselects terminology '#' from the list of terminologies to be packaged.
- (R)emove # - Selects terminology '#' to be removed from the target environments if it exists there.
- (P)ack - Perform the packaging of the selected terminologies.
- (I)lno - Display information from the current property files.
- (B)inaries - Used in S/W development. A tool that deploys the executables. Can be ignored by the user.
- (C)lear - Clears the console by writing a bunch of blank lines.
- (Q)uit - Exits the utility.

Instruction set for "deploy.sh"

- (H)elp - Display a listing of all available commands.
- (DF) DeployFiles
- (DD) DeployDatabase,
- (L)ist - Displays a numbered listing of the terminologies that have been packaged and or the terminologies to be removed. This utility determines the list by reading the contents of the publish directory as pointed to by the value of property "publish_dir" in file "deploy.properties". Please see [EVS:deploy.properties](#) for available deployment properties.
- (I)lno - Display information from the current property files.

- (C)lear - Clears the console by writing a bunch of blank lines.
- (Q)uit - Exits the utility.

Property files use to determine configurations for packaging and deployments.

package.sh: Properties available for package.properties

Property	Description
master_lbconfig	Full path to "lbconfig.props" file. Used as the "source" environment to package from.
publish_dir	Full path to the directory to "publish" to. Needs to be large enough to contain the LexEVS 51 database export files as well Lucene index files for the selected terminologies.

Example file:

```

package.properties

#-----#
# Package Properties:                               #
#-----#

### Lexevs file locations (master)
master_lbconfig=/mnt/lexevs/5.1/resources/config/lbconfig.props
publish_dir=/mnt/lexevs/5.1/lexevsdata51

```

deploy.sh: Properties available for deploy.properties

Property	Description
target_lbconfig	Full path to the "lbconfig.props" file of the target environment to be deployed to.
publish_dir	Full path to the publishing directory.
ssh.key.file	Full path to the SSH private key. A SSH public key needs to be installed under JBoss user account (as normally done with BDA projects).
	<p>The following "target_name_#" properties can be repeated for each web application that needs to be stopped/started after a data deployment. To uniquely identify server properties, target names are appended by a '_#' where '#' is an integer value from 0 to 9 to indicating the server number.</p> <p>Example;</p> <pre>target_url_0=http://localhost:8080/mylexapplication/</pre> <p>Note: Starting deploy.sh with a '-r' option disables stopping and starting of application servers all together.</p>
target_url_#	URL to the lexevsapi51 server. Used by "deploy.sh" to determine if the lexevsapi51 application is running or not.
target_server_#	The application server name.
target_user_#	JBoss user id. Used for the SSH connection needed to stop/start the various application servers.
target_start_jboss_path_#	Full path to JBoss start_jboss script.
target_stop_jboss_path_#	Full path to JBoss stop_jboss script.

Example file:

deploy.properties

```
#-----#
# Deploy Properties:                               #
#-----#

### Lexevs file locations (master)
target_lbconfig=/local/lexevs51/resources/config/lbconfig.props
publish_dir=/mnt/lexevsdata51

### Deployment
ssh.key.file=/local/home/user/ssh-keys/id_dsa_bda

### JBoss Server Information (Max of 10 servers)

target_url_0=http://server:8080/lexevsapi51
target_server_0=server
target_user_0=jbossuser
target_start_jboss_path_0=/usr/local/jboss/server/default/bin/start_jboss
target_stop_jboss_path_0=/usr/local/jboss/server/default/bin/stop_jboss

target_url_1=http://server1:8080/lexevsapi51
target_server_1=server1
target_user_1=jbossuser
target_start_jboss_path_1=/usr/local/jboss/server/default/bin/start_jboss
target_stop_jboss_path_1=/usr/local/jboss/server/default/bin/stop_jboss
```

Downloads

- Utility will be available for download soon.

Install the NCI LexEVS Data Packaging and Deployment Utility

<Work in progress>

Additional Notes

Stopping/starting Jboss

JBoss servers are stopped/started via SSH by invoking the JBoss start_jboss and stop_jboss scripts (as defined in deploy.properties). Up to 10 JBoss servers can be stopped and started this way.
(Stopping/starting of application servers can be disabled using '-r' option on deploy.sh)

Checking if Jboss has started

The deploy app checks the web applications JBoss URL to see if it has started. See [EVS:deploy.properties](#) for details.

Example packaging of terminology:

1. Change to the directory where package.sh has been installed
2. Type package.sh <enter>
3. Type L <enter>
(This lists the terminologies loaded in DEV)
4. Type S # <enter>
(Where # is the number of the terminology in the list)
(This selects the terminology to be packaged)
5. Type p <enter>
(This packs the terminologies selected)
6. Type q <enter>
(Exits the packaging app)

Example database and file system deployment.

1. Change to the directory where deploy.sh has been installed
2. Type `deploy.sh` <enter>
3. Type `L` <enter>
(This lists the contents of the package)
4. Type `dd` <enter>
(This runs the database portion of the deployment)
5. Type `df` <enter>
(This runs the application portion of the deployment)
6. Type `q` <enter>
(Exits the deployment application)