Java Migration Monitor

Users' Guide

Notes and legend: This is a work in progress. It will get more ready together with the JMigMon tool itself. The all the features that are planned but not yet implemented will be seen with yellow background.

The Java Migration Monitor is a tool which helps you to perform and control the unload and load process during the system copy procedure. The Java Migration Monitor is integrated into the SAPinst system copy tool, but it is also possible to use the monitor by starting it manually.
The Java Migration Monitor will

- start the Jload processes to load or unload the data according to user's requirements
- inform the person performing the system copy in case of errors

1. Prerequisites

- JRE version 1.5.0 or higher
- JAVA_HOME environment variable must point to the JRE directory.
- The correct directory structure for Jload dump files must exist on both the source and target hosts

2. Tool

<section consider reviewing>
The tool is located in the JMIGMON.SAR SAPCAR archive.

Content of the archive file:

- migmon.jar; rescheck.jar; activation.jar; mail.jar
- export_monitor.sh / export_monitor.bat
- import_monitor.sh / import_monitor.bat
- res_check.sh / res_check.bat
- export_monitor_cmd.properties
- import_monitor_cmd.properties
- import_dirs.sh / import_dirs.bat
- MigrationMonitor.pdf

3. Configuration

Help

The tool will display the available parameters, if you call it with one of the following command line options:

- -help
- -?

Version

The tool will display the version information (release branch and build date), if you call it with the following command line option:

- -version

General Options

Name	Description	Comments
mode	Migration Monitor mode: import or export	
sid	SID of the target system	Specifies the SID of the target system. This is a mandatory parameter.
dsn	Data Source Name	Specifies the Data Source Name. This is a mandatory parameter.
ssprops	Path to Secure Store properties file	This is a mandatory parameter.
sskey	Path to Secure Store key file	This is a mandatory parameter.
exportDirs	Export directories paths	Specifies path or paths for exported data. To specify multiple directories use ‘;’ on Windows and ‘:’ on Unix as a delimiter. This parameter triggers the export functionality
importDirs	Import directories paths	Specifies path or paths for imported data. To specify multiple directories use ‘;’ on Windows and ‘:’ on Unix as a delimiter. This parameter triggers the import functionality

Additional Options (all optional)

Name	Description	Comments
orderBy	Package order	Can be the ‘name’ or path of the file that contains package names. If the option value is omitted then package order is not determined.
jobNum	number of parallel export jobs	Default is 3.
monitorTimeout	monitor timeout in seconds	Default is 30 seconds.

4. Starting the Migration Monitor

The tool can be started using one of the following:

- The UNIX shell scripts
  jexport_monitor.sh / jimport_monitor.sh
- The Windows batch files
  jexport_monitor.bat / jimport_monitor.bat
- As part of the SAPinst export / import procedure

The application allows you to specify options in the command line and/or in the application property file. The names of the property files are export_monitor.properties and import_monitor.properties. Templates for these files are included in the application archive and must be located in the current user's working directory.

Any options specified in the command line take precedence over the corresponding options in the application property file. Options are case-sensitive; any options that are not recognized are ignored. To specify an option:

- in the command line,
  enter '-optionName optionValue'
- in the application property file,
  insert the new line 'optionName=optionValue'

Example of a command line for a UNIX terminal:

```
./jexport_monitor.sh -sid CE3 -dsn jdbc/pool/CE3 -ssprops
D:/usr/sap/CE3/SYS/global/security/data/SecStore.properties -sskey
D:/usr/sap/CE3/SYS/global/security/data/SecStore.key -exportDirs D:\JPKGCTL
```

Example of a command line for Windows cmd.exe:

```
export_monitor.bat -sid CE3 -dsn jdbc/pool/CE3 -ssprops
D:/usr/sap/CE3/SYS/global/security/data/SecStore.properties -sskey
D:/usr/sap/CE3/SYS/global/security/data/SecStore.key -exportDirs D:\JPKGCTL
```

Start the monitor and close the shell window / command processor. The monitor process will run in background. Use monitor *.log and *.console.log files to check monitor processing state.

Example of an export_monitor.properties file with export options:

```
; jmigmon mode: import or export
mode = export

; number of parallel export jobs, default is 3
jobNum = 1

; SAP SID of the system
sid = CE3

; name of datasource registered in system's SecureStore; normally
jdbc/pool/<SID>
dsn    = jdbc/pool/CE3

; path of the SecureStore properties file
ssprops = D:/usr/sap/CE3/SYS/global/security/data/SecStore.properties

; path of the SecureStore key file
sskey = D:/usr/sap/CE3/SYS/global/security/data/SecStore.key

; list of export directories
exportDirs = D:/JPKGCTL

; monitor timeout in seconds, default is 30
monitorTimeout = 30
```

Example of an import_monitor.properties file with import options:

```
; jmigmon mode: import or export
mode = export

; number of parallel export jobs, default is 3
jobNum = 1

; SAP SID of the system
sid = CE3

; name of datasource registered in system's SecureStore; normally
jdbc/pool/<SID>
dsn   = jdbc/pool/CE3

; path of the SecureStore properties file
ssprops = D:/usr/sap/CE3/SYS/global/security/data/SecStore.properties

; path of the SecureStore key file
sskey = D:/usr/sap/CE3/SYS/global/security/data/SecStore.key

; list of import directories
importDirs = D:/export/unpacked/JAVA/JDMP

; monitor timeout in seconds, default is 30
monitorTimeout = 30
```

What happens during export / import:

While importing the tool starts search in the directories specified by -importDirs parameter for packages in XML format and put them in a working queue. Next it starts a number (specified by -jobNum parameter) of parallel Jload importing tasks, taking tasks from the working queue until the queue is empty.

While exporting the tool starts search in the directories specified by -exportDirs parameter for packages in XML format and put them in a working queue. Then it starts exporting all the packages containing meta data one after another (not in parallel) while removing them from the queue. The tool then starts a number (specified by -jobNum parameter) of parallel Jload exporting tasks, taking tasks from the working queue until the queue is empty.

ToDo

5. Output Files

Export

- export_monitor.log
- export_state.properties
- ExportMonitor.console.log

Import

- import_monitor.log
- import_state.properties
- ImportMonitor.console.log

Both the export and import state files contain package state lines such as the following:
SAPUSER=+

Format of lines is <PACKAGE>=<STATE>. Possible values for state are:

0	Package export/import not yet started.
?	Package export/import in progress.
-	Package export/import finished with errors.
+	Package export/import finished successfully.

If any ftp/net exchange options are used, then the export state file may contain a second <STATE> column, which refers to the state of the package transfer.

Then the export state file contains package state lines such as the following:
SAPUSER=++

Format of lines is <PACKAGE>=<STATE>. Possible values for state are:

0	Package export not yet started.
?	Package export in progress.
-	Package export finished with errors.
+0	Package export finished successfully; package transfer not yet started.
+?	Package transfer in progress.
+-	Package transfer finished with errors.
++	Package transfer finished successfully.

6. Restarting Jload Processes

The state file allows package states to be manually updated to restart failed Jload processes.
For example, if package processing failed and the package state has the value –, the state can be set to 0 and processing of the package will be started again.

To restart package processing, set the package state from - to 0.
To skip package processing, set the package state from 0 or - to +.
(This is not recommended because it can cause inconsistent data files or database content.)
If the package is currently being processed (the package state is ?) then any manual modifications of the package state are ignored.

7. Release Notes

New Features in VERSION X

- ToDo Some new features here