EPV for z/OS+

Migration tool

Version 11

EPV Technologies

January 2013

INDEX

1. Introduction

This document describes the steps that must be performed to upgrade the EPV for z/OS+ product (**EPV**) to version 11.
At this scope EPV Technologies provides an useful tool that you can find inside the installation cd under the directory:

- EPVROOT/PRODUCTS/EPVZOS_V11/MIGRATION

This tool allows you to migrate your data from your old environment to the new one, even if you have decided to use another database engine. If you decide to use the same database engine you can choose to rename the old database names by adding a suffix (es.: mdetaold, mconfold, mresaold etc.) while maintaining the old names for the new databases or build a new set of databases using a different standard for their names (es: mdeta11, mconf11, mresa11 etc.). If you are using MySQL we advise you to opt for the second choice due to the complexity of the database rename operation.
Anyway, if you decide to rename the old databases before performing the migration step, you can find some useful code, for all database engines supported by EPV, inside the installation cd under the directory:

- EPVROOT/PRODUCTS/EPVZOS_V11/MIGRATION/RENAME DB

2. Migration tool

Inside the EPV installation cd, under the EPVROOT/PRODUCTS/EPVZOS_V11/MIGRATION directory you can find the EPV Migration tool, it includes the following customizable members:

1. EPVMIGRATION.exe is the executable member that perform the migration step
2. MIGRCONFIG.PL is the configuration file.
3. INPUTCONN.PL is the user exit that contain the input database information.
4. OUTPUTCONN.PL is the user exit that contain the output database information.
5. EXECMIG.PL is the user exit that manage the databases migration.

2.1. EPVMIGRATION

After you have customized the configuration file and the databases user exit you have to run this program. It performs the migration steps which copy the data from the old databases to txt files created inside the work directory specified in the configuration member and then creates and loads the files in the new database structures.

2.2. MIGRCONFIG

Is the configuration file of the Migration tool, it must be customized by setting the value for the following variables:

1. $INPVER is the version of the old installation of EPV. You can retrieve this information inside the $MAINT member located under the EPVROOT directory.
2. $OUTVER is the version of the new installation of EPV.
3. $DB_INPUT is the input database engine (MYSQL or SQLSERVER).
4. $DB_OUTPUT is the output database engine (MYSQL or SQLSERVER).
5. $CREATE_TABLES flag used to create the new tables or append the migrated data to tables already created.
6. $CREATE_INDEX flag used to create or not the index on the final tables.
7. $USE_DUMP flag used to migrate data using txt files or not. When you don't change the db engine it's better to set this parameter to 'N'
8. $DEBUG is the logs detail level 0-2
9. $WORKDIR is the work directory in witch the Migration tool perform the dump of the old databases. You must be sure that this directory is visible and accessible to all databases involved in the migration step.
10. $LOGPATH is the path in witch the Migration tool creates the log.

2.3. INPUTCONN

This file contains all the information related to the input database.
At the beginning of the file you can find a set of variables that must be customized with the names of the input databases of your environment:

```
our %inDbs = (WORK  => 'work',
              MDETA => 'mdeta',
              MCONF => 'mconf',
              MRESA => 'mresa',
              MWKLA => 'mwkla',
              MWRMF => 'mwrmf',
              MSTAT => 'mstat',
              MTRND => 'mtrnd');
```

if you decide to use the same database engine and to rename the old databases before the migration step, you have to specify here the new names chosen.
After you have setted the described variables you have to customize the database connection. Inside the INPUTCONN.pl member you’ll find a section for each database engine supported by EPV (MYSQL, SQLSERVER), choose the one that reflects your environment and customize the “host”, “user” and “password” information for the connection included.

2.4. OUTPUTCONN

This file contains all the information related to the output database.
At the beginning of the file you can find a set of variables that must be customized with the names of the output databases of your environment:

```
our %outDbs = (WORK  => 'work',
               MDETA => 'mdeta',
               MCONF => 'mconf',
               MRESA => 'mresa',
               MWKLA => 'mwkla',
               MWRMF => 'mwrmf',
               MSTAT => 'mstat',
               MTRND => 'mtrnd');
```

if you decide to use the same database engine, the names of the new databases must be different from the old ones.
After you have setted the described variables you have to customize the database connection. Inside the OUTPUTCONN.pl member you’ll find a section for each database engine supported by EPV (MYSQL, SQLSERVER), choose the one that reflects your environment and customize the “host”, “user” and “password” information for the connection included.

2.5. EXECMIG

This user exit contains a flag for each database used by EPV. Setting to 1 the value of a flag you decide to do the migration for the related database.
This is a useful exit if you want to test the migration tool on one database before performing the entire migration step.