Procedure Documentation

Generic data migration procedure for SAP Hybris 6.2 in Distrelec upgrade project

30th January 2017

Version 1.0

Prepared by: Lukasz Nowakowski

Custom item types and relations included in migration procedure:

|  |  |
| --- | --- |
| **internalcode** | **itemtypecode** |
| DistErpPriceConditionType | 20064 |
| DistCarpetContentTeaser | 20087 |
| DistCarpetContentTeaserWithText | 20088 |
| DistCarpetItem | 20089 |
| DistExtCarpetItem | 20092 |
| DistExtHeroRotatingTeaser | 20093 |
| DistExtHeroRotatingTeaserItem | 20094 |
| DistHeroRotatingTeaser | 20095 |
| DistHeroRotatingTeaserItem | 20096 |
| DistHeroRotatingTeaserWithText | 20097 |
| DistRmaRequestProcessEntry | 20065 |
| BaseStore2RegCountryRel | 20053 |
| DistDownloadMedia2Language | 20063 |
| DistVideoMedia2Language | 20066 |

Procedure steps:

1. Validate item type codes for deployments according to *distrelecB2Bcore-items.xml* configuration
2. Count generic items and relations, which will be migrated. Execute and keep results of SQL scripts:
   1. generic items: 0.count\_genericitems.sql
   2. relations: 0.count\_links.sql
3. Check existence of ‘*TypeSystemName*’ column in ‘*ydeployments*’ table in database (column will be created automatically during platform 6.2 startup)
4. Ensure that SAP Hybris platform is shut down
5. Ensure sys\_master media(H4.8) folder back up is complete
6. Ensure hybris database(H4.8) back up is complete
7. Depending on the outcome of test from step 3. execute SQL:
   1. if type system column exists: migration-script\_pass\_1a.sql
   2. if there is no type system column: migration-script\_pass\_1b.sql

**We proceed with the deployment of v6.0.0**

1. Execute SQL: migration-script\_pass\_2.sql

**6.1** We should disable the task-service before starting Hybris using the following properties

cronjob.timertask.loadonstartup=false

task.engine.loadonstartup=false

1. Start SAP Hybris platform
2. Execute type system clean-up in HAC

*(Note: This step is optional and can be either skipped or executed without “remove instances” option checked)*

**8.1** Run the SQL script **migration-script\_pre\_system\_update.sql** before starting the next step.

**8.2** Drop the table « ISHOP\_CUSTOMER\_EXPORT » which contains unknown column types

**8.3** remove from the table « ATTRIBUTEDESCRIPTORS » the following records:

'heroRotatingTeaserPOS'/‘p\_herorotatingteaserpos’ for the hero rotating teaser

'subText/p\_subtext’  for both hero rotating teaser and DistCarpetContentTeaser

’text/p\_text’  for both hero rotating teaser and DistCarpetContentTeaser

’title/p\_title’  for both hero rotating teaser and DistCarpetContentTeaser

1. Execute type system update (with all required settings) in HAC

Import the CSV file « /reports/oracle/reports.csv » after the system update.

1. Execute SQL: migration-script\_pass\_3.sql (optional, removing obsolete deployments)
2. Validate database table used to store type contents to be different than *‘genericitems*’ in hMC (System → Types → *find and edit one of the generic types* → Administration → Table)
3. Execute SQL: 1.set-to-generics.sql
4. Execute in HAC: reload-type-system.groovy
5. Validate database table used to store type contents to be ‘*genericitems*’ in hMC (“Reload” in edit mode is sufficient) (Example: run the following flexible search query and ensure that the generated SQL query is getting the data from genercitems table « select {pk} from {DistErpPriceConditionType} »)
6. If step 16. failed, return to step 15.
7. Export generic data and relations using ImpEx: type-system-migration\_distrelec-generics\_no-createts.impex in hMC (System → Tools → Export)
8. Validate exported data package (for example compare CSV lines count with results of step 2.)

**19.1** Save the PKs of the current ErpPriceConditionTypes for later usage.

1. Import ImpEx in HAC: delete-generics.impex
2. Execute SQL: 2.set-to-deployments.sql
3. Execute in HAC: reload-type-system.groovy
4. Perform test from step 22. Results should be equal. If not, return to step 20.
5. Import data exported in step 18. in hMC (System → Tools → Import)
6. Validate imported data contents with values gathered in step 2. (for example compare ‘*cnt*’ value with COUNT() on equivalent database table)
7. Update the medias using the groovy script media-update.groovy

**26.1 remove all medias from the distrelecProductCatalog using the groovy script *remove\_medias.groovy***

**this script will take very long time ( > 8h ). It has to be executed before the Master PIM import.**

1. Before starting a new Master PIM import, reset the hashes of the Products and Media containers:

UPDATE mediacontainer set P\_PIMXMLHASHMASTER = NULL;

commit;

UPDATE productslp SET p\_pimXmlHashLocalized = null WHERE langpk IN (SELECT pk FROM

languages lang WHERE lang.isocode='en');

commit;

delete from listdidome2pr where PK is not null;

commit;

delete from ListDiViMe2Pr where PK is not null;

commit;

Update the UID of the root navigation nodes from SiteRootNode to ROOT:

update cmsnavigationnode item\_t0 set item\_t0.p\_uid='ROOT' WHERE ( item\_t0.p\_parent IS NULL and item\_t0.p\_uid ='SiteRootNode');

commit;

Use direct SQL to update customers having old password encoding that are not supported anymore such sha1\_elfa and md5\_ishop

UPDATE users item\_t0 set item\_t0.p\_active=0 WHERE item\_t0.Encode not in ('\*', 'plain', 'md5', 'sha\_256', 'sha-256', 'sha-512', 'pbkdf2');

UPDATE users item\_t0 set item\_t0.Encode='sha\_256' WHERE item\_t0.Encode not in ('\*', 'plain', 'md5', 'sha\_256', 'sha-256', 'sha-512', 'pbkdf2');

commit;

Remove the *ELFA* and *MOVEX* enum values from the database.

Use direct SQL query to update the prices with the new ErpPriceConditionType PKs, something like

update PRICEROWS set P\_ERPPRICECONDITIONTYPE='8796093140576' where P\_ERPPRICECONDITIONTYPE='8800484065379';

update PRICEROWS set P\_ERPPRICECONDITIONTYPE='8796093075040' where P\_ERPPRICECONDITIONTYPE='8800483999843';

update PRICEROWS set P\_ERPPRICECONDITIONTYPE='8796093107808' where P\_ERPPRICECONDITIONTYPE='8800484032611';

commit;