**Background:**

**nxmgr\_generate\_appuid\_upgrade\_xml**

**nxmgr\_upgrade\_appuid**

We have come across some cases where NX datasets created/saved in very old versions of NX (NX2 and earlier), not having APPUID-OBJ named references in the non-master datasets which was causing problems when newer versions of NX worked with the data. The above two Teamcenter utility programs can be used to repair NX Non-Master (UGPART, UGALTREP, UGSCENARIO) type datasets that are missing the “APPUID-OBJ” named reference object in the database. When upgrading from NX2 or prior releases to NX4 and above, it’s advised that the customers run this utility after upgrading the database (if Teamcenter version being upgraded as well), prior to using newer version of NX.

So typical steps would be:

1. Upgrade the database to newer Teamcenter release
2. Run **nxmgr\_generate\_appuid\_upgrade\_xml and nxmgr\_upgrade\_appuid** utilities to create the missing APPUID-OBJ references
3. Run NX refile (if desired; this is not required).

These two utilities must be run as an administrator and nxmgr\_upgrade\_appuid should be run in bypass mode.

NOTE: Although any NX non-master dataset may potentially be missing an APPUID-OBJ named reference, this condition is only known to cause problems for UGALTREP datasets. For this reason the document below assumes that it is only UGALTREP datasets that are being dealt with. (If for any reason you want to examine other non-master type datasets

**Install Instructions:**

* Find your Teamcenter installation’s %TC\_ROOT%\bin. Make backup copies of the files nxmgr\_upgrade\_appuid.exe and libnxmgr\_im.dll located there.
* Copy the exe and dll files in the attached zip file to the %TC\_ROOT%\bin directory. Remove “\_rename” from the exe file “nxmgr\_upgrade\_appuid.exe\_rename”.

**Recommendation:**

Before running this utility on the entire database, take some sample data where you are having problems and run the utility the on the problematic items, make sure that your issues are fixed. Then you can run it on the whole database.

**Execution Instructions:**

To use these programs, follow the following steps:

1. First, you will need to determine what items in the database need to be analyzed to see if they contain datasets whose APPUID-OBJ named references need restoring. This can probably be done most easily through a Teamcenter query to find all items containing item revisions having any non-master (UGPART, UGALTREP and UGSCENARIO) datasets. It is ok to run this utility on items containing datasets that already contain the APPUID-OBJ reference. It will ignore such items.
   1. Alternately, if you know of particular non-master datasets that are missing their APPUID-OBJ named reference you can target just those datasets. This may be useful as a first test before running this procedure on a larger set of parts.
   2. It is possible to get an exact list of all non-master datasets in a database that are missing the APPUID-OBJ named reference, but the procedure is rather complicated. See the Appendix at the end of this document for more information on how this might be done.
2. These items should be put into a text file, one per line, in the standard “DB” style format used to refer to items in the database. For example, your file would look like this:

target\_items.txt

@DB/012345

@DB/bolt1

@DB/hinge2

1. Now run the **nxmgr\_generate\_appuid\_upgrade\_xml** programs to generate an xml file that will be used in a later step to actually perform the upgrade. To do this, go to a Teamcenter command prompt window and run the nxmgr\_generate\_appuid\_upgrade\_xml command, using the following command line arguments:

* –u=<user>
* -p=<pwd>
* -g=<group>
* -input\_file=<text file you created in step 2 above>
* -xml\_file=<name of output file to be written by this utility>
* -upgrade\_released=<yes|no> (default is no – decides if released parts should included)
* -bypass=<yes|n> (default is no – turns on bypass privilege, **has to be set to yes if upgrade\_released=yes is to be used)**

You must specify both the input file and an output xml file.

Example:

*nxmgr\_generate\_appuid\_upgrade\_xml.exe -i=target\_items.txt -xml\_file=input.xml - -upgrade\_release=yes –bypass=yes -u=infodba -p=infodba*

This utility validates the existence of items listed in the target\_items.txt and generates an XML file containing those items.

1. The output file created in step 3 (input.xml) should contain the items you provided in a simple xml-style format.
2. Now you’re ready to run the main upgrade program. This program needs to be run by an administrator user who has bypass privilege. No other users should be logged into the database while this utility is being run.
3. Once you’re ready, run the nxmgr\_upgrade\_appuid command in the Teamcenter command prompt window using the following arguments:

* –u=<user>
* -p=<pwd>
* -g=<group>
* -bypass=yes
* -xml\_file=<the xml file you generated in step 3 above>
* Several other optional arguments (see below)

The nxmgr\_upgrade\_appuid utility will produce three output files; by default these files will be located in the same directory as the input xml file and will have following names:

* <xml\_file\_name>.auid.log – This a report log that shows what items, revisions, and datasets were processed by the utility during the run
* <xml\_file\_name>.auid\_conflicts.log – This is a “conflict” log; it records the names of parts that had irregular APPUID data that could not be corrected by this utility (for example, UGALTREP datasets whose APPUID objects do not match across revisions or datasets containing more than one APPUID object.)
* <xml\_file\_name>.auid\_err.log – This is an error log that contains messages about any parts that encountered errors and could not be processed by the utility.

Example:

*nxmgr\_upgrade\_appuid -u=infodba -p=infodba -g=dba -bypass=yes -update\_mod\_props=no -xml\_file=input.xml -remove=no*

Running the utility will also produce a syslog of the kind you would get by running any ITK program; this will be located in the system temporary directory.

If you want to specify different locations for these files than the ones shown above you can do so with the following optional command line arguments:

* -report=<full path to desired output report file>
* -conflict\_report=<full path to desired output conflict report file>
* -errlog=<full path to desired output error log>

Other optional command line arguments are these (default values are shown in CAPS):

* -dryrun=<NO|yes>
  + If yes, generate the output files and logs but do not actually create or modify any objects in the database
* -debug=<NO|yes>
  + If yes, generates extra debugging information to the syslog.
* -help
  + If passed in then prints out usage information to the console
* -remove=<NO|yes>
  + If “yes”, this option removes the APPUID-OBJ named references if they are found to be incorrect. **It’s not needed from the upgrade/repair perspective and should be left unset (so it defaults to NO) unless you have a specific reason to do otherwise.**
* -update\_mod\_props=<YES|no>
  + If “no”, it will not modify the last\_modified\_date property of the datasets that are actually modified by the utility.

Appendix – How to find datasets with missing APPUID-OBJ named references

Simple, less efficient method:

The simplest way to generate an input list of items to use in step 2 above is simply to run a query that will find all items in the database that have one or more item revisions containing UGALTREP datasets. (The text of this query, along with two other queries, is contained given at the very end of this document.) You can import this query into your database using the Query Builder application in Teamcenter.

Once you have the query imported, you just have to execute it with no extra input (leave the dataset type input set to “UGALTREP”). The resulting list of items produced by this query will be all items that have one or more UGALTREP datasets in any of their item revisions. You can use this list to construct the input to the generate xml utility described in step 2 near the top of this document.

Note that many, probably most, of the items on this list will contain only altrep datasets that already have their APPUID-OBJ named references and so do not need repairing. ***This is OK, since running the upgrade utility will have no effect on such datasets***. The only disadvantage of doing things this way is that the utility may take much longer to run than it would on a smaller, more refined list if input items. (Of course, this will depend on how many UGALTREP datasets you have in your database and how many are missing their APPUID-OBJ named references.)

If you don’t want to do this, there is another way that involves considerably more manual work but will produce a more restricted input list for the upgrade utility.

More complex but more efficient method:

Because of the limitations of the Teamcenter QueryBuilder application, the process of using Teamcenter queries to exactly identify those items (and ONLY those items) that contain UGALTREP datasets needing repair is unfortunately rather complicated.

There are two queries included in this document that may be helpful in this process. (These are the second and third queries whose text is below.) The first of these can be used to find at least some of the Items in the database that contain datasets with missing APPUID named references. The second can be used to directly find the datasets that are missing these named references. Using both of these queries in succession will allow you to find all the datasets that need to be repaired using the nxmgr\_upgrade\_appuid utility.

To import these queries into your database, copy the text for each on into its own text file, and then use the Import function in the QueryBuilder application. You can change the name and description of each query as you wish.

Using the queries:

The provided Item-based query can be used to find all items in the database such that ALL of the UGALTREP datasets contained in any revisions of that item are missing their APPUID-OBJ named references. **Note that this will not find items that contain some UGALTREP datasets with missing APPUID-OBJ named references and also contain “correct” UGALTREP datasets that have all their named references.** In most cases this will identify many but not all of the problem datasets in the database, but is probably the best first step. The second query, described below, can then be used to identify any remaining datasets that need to be repaired.

To use the Item-based query it will be necessary to run it twice and compare the results.

The first time, run the query with the following input fields set:

Dataset\_type:  UGALTREP

Ref\_names:  APPUID-OBJ

The output from this query will be all Items in the database that have one or more Item Revisions that contain UGALTREP datasets that HAVE a named reference of the relevant type (APPUID-OBJ).  For the second run, execute the query with the ref\_named input field blank (leave the dataset\_type set to UGALTREP), as follows:

Dataset\_type:  UGALTREP

Ref\_names:

This output of this run will be Items in the database that have Item Revisions that contain any UGALTREP datasets (both those with APPUID-OBJ named references and those without).

**The Items that appear in the second query results but not in the first will be the ones you’re looking for and should be the ones used to construct the initial input list of Items in step 2 above.**

The easiest way to make the comparison between the two sets of results is probably to display the results of each query in the “Details” tab, sort them by name (or whatever way you prefer) and then use one of the buttons there to export the results to Word or to Excel.  Excel is probably better because I suspect there are probably some comparison functions of some kind in there that you might be able to use to compare two sheets and have it tell you what rows appear on one but not the other.

If you have run the appuid upgrade process on the set of items you just identified, there may remain some UGALTREP datasets in the database that still need to be repaired. The second query can be used to identify these datasets (if any exist).

The second query provided here will find all the UGALTREP datasets themselves are missing their APPUID-OBJ named references. To do this, you will need to run the query twice and compare the results in the same way as described for the Item-based query just above (the inputs are the same).

**The datasets that appear in the second query results but not in the first will be those that are still missing their APPUID-OBJ named references. If the number of datasets in the two queries is the same then no datasets remain that need to be repaired.**

Since this is a dataset-based query it will be necessary to determine the items to which these datasets belong. In some cases it may be possible to deduce the owning item from the dataset name, depending in your naming conventions. In other cases it may be necessary to find the references of the dataset in the Teamcenter UI.

Text of queries:

This is the basic query to find all items in the database containing altrep datasets:

<?xml version="1.0" encoding="UTF-8" ?>

<plmxml\_bus:PLMXMLBusinessTypes xmlns:plmxml\_bus="http://www.plmxml.org/Schemas/PLMXMLBusinessSchema" schemaVersion="4" date="2003-10-08" time="13:09:29" author="unset">

<plmxml\_bus:SavedQueryDef id="id1" name="All Items with UGALTREPs" nameRef="id2" descriptionTextRef="id3" queryFlag="0" queryClass="Item" resultsType="0" >

<Description>finds all items in the database that have at least one revision containing one or more datasets of type UGALTREP</Description>

<plmxml\_bus:QueryClause stringValue="SELECT qid FROM Item WHERE &quot;ItemRevision&lt;-items\_tag.Dataset:IMAN\_UG\_altrep.dataset\_type.datasettype\_name&quot; = &quot;${DatasetType = UGALTREP}&quot;"></plmxml\_bus:QueryClause>

</plmxml\_bus:SavedQueryDef>

<Text id="id2" primary="en\_US">

<Item language="en\_US">All Items with UGALTREPs</Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

</Text>

<Text id="id3" primary="en\_US">

<Item language="en\_US">finds all items in the database that have at least one revision containing one or more datasets of type UGALTREP</Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

</Text>

</plmxml\_bus:PLMXMLBusinessTypes>

This is the second, more complex Item query described above:

<?xml version="1.0" encoding="UTF-8" ?>

<plmxml\_bus:PLMXMLBusinessTypes xmlns:plmxml\_bus="http://www.plmxml.org/Schemas/PLMXMLBusinessSchema" schemaVersion="4" date="2003-10-08" time="13:09:29" author="unset">

<plmxml\_bus:SavedQueryDef id="id1" name="APPUID - Items" nameRef="id2" descriptionTextRef="id3" queryFlag="0" queryClass="Item" resultsType="0" >

<Description></Description>

<plmxml\_bus:QueryClause stringValue="SELECT qid FROM Item WHERE &quot;ItemRevision&lt;-items\_tag.Dataset:IMAN\_UG\_altrep.dataset\_type.datasettype\_name&quot; = &quot;${DatasetType = UGALTREP}&quot; AND &quot;ItemRevision&lt;-items\_tag.Dataset:IMAN\_UG\_altrep.ref\_names&quot; = &quot;${Reference = APPUID-OBJ}&quot;"></plmxml\_bus:QueryClause>

</plmxml\_bus:SavedQueryDef>

<Text id="id2" primary="en\_US">

<Item language="en\_US">Appuids</Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

<Item language="pt\_BR"></Item>

</Text>

<Text id="id3" primary="en\_US">

<Item language="en\_US"></Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

<Item language="pt\_BR"></Item>

</Text>

</plmxml\_bus:PLMXMLBusinessTypes>

This is the Dataset query:

<?xml version="1.0" encoding="UTF-8" ?>

<plmxml\_bus:PLMXMLBusinessTypes xmlns:plmxml\_bus="http://www.plmxml.org/Schemas/PLMXMLBusinessSchema" schemaVersion="4" date="2003-10-08" time="13:09:29" author="unset">

<plmxml\_bus:SavedQueryDef id="id1" name="APPUID - Dataset" nameRef="id2" descriptionTextRef="id3" queryFlag="0" queryClass="Dataset" resultsType="0" >

<Description>Find UGALTREP datasets with APPUID-OBJ named ref</Description>

<plmxml\_bus:QueryClause stringValue="SELECT qid FROM Dataset WHERE &quot;dataset\_type.datasettype\_name&quot; = &quot;${dataset type = UGALTREP}&quot; AND &quot;ref\_names&quot; = &quot;${ref\_names = APPUID-OBJ}&quot;"></plmxml\_bus:QueryClause>

</plmxml\_bus:SavedQueryDef>

<Text id="id2" primary="en\_US">

<Item language="en\_US">Appuid Finder</Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

<Item language="pt\_BR"></Item>

</Text>

<Text id="id3" primary="en\_US">

<Item language="en\_US">Find UGALTREP datasets with APPUID-OBJ named ref</Item>

<Item language="fr\_FR"></Item>

<Item language="de\_DE"></Item>

<Item language="it\_IT"></Item>

<Item language="es\_ES"></Item>

<Item language="pt\_BR"></Item>

</Text>

</plmxml\_bus:PLMXMLBusinessTypes>