Background:

nxmgr_generate_appuid_upgrade_xml
nxmgr_upgrade_appuid

These 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. NX datasets created/saved in very old versions of NX (NX2 and below) will not have APPUID-OBJ named references in the non-master datasets and this will cause problems when newer versions of NX works with data. When upgrading from NX2 or prior releases to NX4 and above, customers must 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

These two utilities must be run as an administrator and nxmgr_upgrade_appuid should be run in bypass mode.

Install Instructions:

Remove "_rename" from the attached exe file "nxmgr_upgrade_appuid.exe_rename"
Copy the.exe file to %TC_ROOT%\bin directory.

Versions of the Software:

Teamcenter version: TC 8.3.2
Platform: Windows 32 bit

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 the items that already contain the APPUID-OBJ reference. It will ignore such items.

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

3. 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>

   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 -format=compact -upgrade_release=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.

4. The output file created in step 3 (input.xml) should contain the items you provided in a simple xml-style format.

5. 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.

6. 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 found to be incorrect. It's not needed from the upgrade perspective.
- -update_mod_props=<YES|no>
  - If "no", it will not modify the last_modified_date property of the impacted datasets