

Network Manager

Release Notes v4.5.0.0 Fix 20

Introduction

This document defines the changes made to the Network Manager product for fix release v4.5.0.0 Fix 20 and is specifically targeted at end users.

After reading through this document, should you have any further training or consultancy requirements then please contact your **exor** account manager.

Fix Details

Baseline Release	4.5.0.0
Fix Description	Failure of the originally released upgrade scripts designed to upgrade from 4400 to 4500 results in view definitions that are not supported by the pragma. The fix is to repair any failings in the upgrade process. See below for details.
Prerequisites	
Implementation Instructions	<p>Unzip nm_4500_fix20.zip to a staging folder.</p> <p>Then copy in the new version of the following files into the given folders from the staging folder and copy each of the files into the named folder.</p> <p>Then proceed with the documented upgrade method as described in the upgrade guide or, if already upgraded, follow the procedure below.</p>
Limitations	The fix should be applied before any other fixes to 4.5.0.0 or as soon as possible thereafter. The code in this fix contains a new version of nm3sdm package body which will be undone by fix 11. Do not apply fix 11 after fix 20. If fix 11 has already been applied, the fix 20 includes all of its components and no problems will arise.
Configuration Information	None
How To Test	Recommend full regression test
Rollback Strategy	Initially implement on a test environment

List of Amended Files

Filename	Version	Destination Folder
nm3sdm.pkw	2.59	nm3\admin\pck
regen.sql	3.4	nm3\install
nm4400_nm4500_metadata_upg.sql	3.4	nm3\install
nm4400_nm4500_upg.sql	3.9	nm3\install
nm_4500_fix20.sql		
remove_private_views.sql		nm3\install
v_nm_rebuild_all_inv_sdo_join.vw	3.4	nm3\admin\views
v_nm_rebuild_all_nat_sdo_join.vw	3.3	nm3\admin\views

Upgrade failure – Summary

This chapter summarises all reasons why the original upgrade to release 4.5.0.0 had the potential of some failure and the software changes that have been made in this fix release to improve the upgrade process. If the upgrade has been performed on an existing schema, it is important that the following chapter is understood and that the second upgrade log file is inspected for failures such as those identified. If the fix is being applied prior to upgrade, the new configuration of scripts will not deliver these failures. If an upgrade has already been performed issues may not arise and remain hidden, however, the repair script should be executed.

The failure can arise in the attempt of the original upgrade to perform the following:

- Removal of all subordinate-user private views. These were included in all previous releases to allow third-party products to work with subordinate user accounts. The 4.5 release now uses the more usual method of standard synonyms (usually public synonyms) as indicated from the option HIGPUBSYN. This saves a great deal of execution time in processing view definitions for a theme/subordinate user when theme roles are granted and provides significant improvement in performance during generation of theme data.
- Removal of references to NM3CONTEXT.GET_EFFECTIVE_DATE. This function uses some complex PL/SQL to deliver namespaces and context variable values. Its replacement with the newer method of context variables held in the namespaces under the EXOR_CORE schema provides performance improvements by removing the context switching between PL/SQL and SQL. This is highly desirable, particularly in spatial views. Attempts are also made to remove the reference to this function inside triggers that exist on inventory views.
- Any view definitions with a retained dependency on NM3CONTEXT such as the function GET_EFFECTIVE_DATE will fail when used in a theme in locator. The NM3CONTEXT has had some pragma definitions removed from the package specification. As a result, the code used to check if a specific record exists with a theme will fail with an error such as:

```
ORA-20000: HIG-0150: GIS Theme data is invalid:

LINEAR GROUP_DT
statement violates WNDS RESTRICT_REFERENCES pragma
ORA-06512: at "HIGHWAYS.HIG", line 1464
ORA-06512: at "HIGHWAYS.HIGGIS", line 1008
```

This will manifest itself as a failure to zoom to selected features in the locator form/map combination.

- Inventory view triggers may fail to be refreshed. Any such failures will be marked “Failed to create trigger for inv type: XXXX” inside the log file. This is almost certainly due to problems in the metadata. Users should refresh the view definitions for any asset types listed in this section and the structure of the metadata should be confirmed to be correct and consistent with the view definition.
- If the system upgrades are executed more than once, context values may already be instantiated and this can lead to an error being reported as shown below.

```
Re-initialising Context...
BEGIN
*
ERROR at line 1:
ORA-20498: Cannot set context values for read only attributes
ORA-06512: at "ATLAS.NM3CONTEXT", line 290
ORA-06512: at line 3
```

This has no impact on the upgrade and can be ignored.

Failure Details

The upgrade process had the potential of failure due to errors in:

- Order of execution of individual scripts in the upgrade.
- Individual script failure

Any conflict arising out of retained subordinate-user views will likely remain unnoticed and no issues should arise. The upgrade process is described below with some indicators marked where there is a potential for failure. Some customer sites that have already upgraded may have experienced one or more of the problems. Some, such as system test environments, showed no sign of error. If there are any concerns about an upgrade that has been

performed with the original install files, a repair script is provided. Please note that in some exceptional circumstances, execution of the original upgrade files would force view definitions and their dependencies such as instead-of triggers to be refreshed from their original scripts.

The upgrade is performed with a main driving script named nm4400_nm4500_upg.sql. Inside this script a call is made to a file named regen.sql under the header

```
--      ***** REGENERATE SYSTEM BUILT OBJECTS *****
```

The original regen.sql script is executed virtually immediately after the recompilation of all objects. Hence few (if any) views would be in an invalid state. The script initially performs a PL/SQL anonymous block which attempts to address spatial view definitions that have been generated as asset and road-group themes through the use of standard product APIs. The original code has no exception handling and as a result of a failure some of the intended views will fail to be refreshed. Failure can arise as a result of:

- View definitions are updated and synonyms are then refreshed. If private synonyms are employed the synonym-refresh will fail as conflict exists between the new subordinate-user synonym and the view definitions that would be removed later in the process.
- View definitions of a standard naming convention linked to a standard theme may be modified in a non-standard way – for example NSG AD data is an asset type and views holding their spatial representations adopt the same naming convention as standard asset views but their structure is modified for use with the NSG product.

Any such failure will be represented in the log file under the heading “Regenerate system built objects with new context functionality” and the full SQL that has failed to execute will be displayed. This would cause the anonymous block of PL/SQL to abort leaving some standard view definition to retain their dependency on NM3CONTEXT.GET_EFFECTIVE date. This can then cause the HIG-0150 exception during use of Locator.

The original regen.sql file would then make an attempt to address all invalid views, swapping the references to NM3CONTEXT.GET_EFFECTIVE_DATE with a reference to a sys-context variable. The driving cursor was failing in several ways. Firstly, executed directly after a full schema compile-all, few of the views would be invalid and secondly, the views that were affected (if any) would be refreshed and the method used could invalidate the view and/or remove instead-of triggers. If you have executed an upgrade using the original upgrade files then please check view definitions, ensuring any instead-of triggers are present and comments are also present. As stated, it is highly unlikely that any views were affected since all views would likely be in a valid state after the compile all. Use the list of invalid objects in the log file under the banner “List of Compilation Errors (if any)” to see if any are invalid after the completion of the compilation.

The regen file then moves on to remove dependencies on NM3CONTEXT functions by refreshing inventory views and triggers using the standard product APIs. As stated above, this could fail and any such failures will be marked “Failed to create trigger for inv type: XXXX” inside the log file. This is almost certainly due to problems in the metadata. Users should refresh the view definitions for any asset types listed in this section and the structure of the metadata should be confirmed to be correct and consistent with the view definition.

After completion of the regen.sql script, the main upgrade script will fire the nm4400_nm_4500_metadata_upg.sql script which will make an attempt to remove subordinate-user views. This script will remove all synonyms and views for subordinate users and is prone to fail due to buffer-overflow.

The nm4400_nm4500_metadata_upg.sql will then complete but the main upgrade script completes with all synonyms being refreshed according to the HIGPUBSYN option. In the case of a system being configured with public synonyms, it is highly unlikely that any problem in relation to the subordinate user access to objects would arise since any missing synonyms would be created but no conflict would exist. In the case of using private synonyms, again, no problems are likely to be encountered as subordinate users would still gain access through their own private views.

Failure Resolution

The fix is in two parts. The first part addresses an upgrade yet to be applied and the second part addresses upgrades that have been performed by supplying a repair script.

Fix for upgrades yet to be applied

For fixes to be upgraded, unzip the files from fix release zip file into a staging folder. Copy the files into the folders as shown in the list of amended files above.

Execute the upgrade as in the upgrade guide. The changes to the amended files and the order of execution will prevent upgrade failures. At the close, no subordinate user should have any view definitions for the spatial themes in their own schema. Also, all view definitions that were valid, that had no instead-of triggers and had a first-level dependency on NM3CONTEXT.GET_EFFECTIVE_DATE should have been changed with the reference now being made to SYS_CONTEXT('NM3CORE', 'EFFECTIVE_DATE').

The modified scripts will post a record in the table of applied upgrades to inform the system that nm_4500_fix20 has been applied.

As in any upgrade, the log files should be inspected. Contact support if any unexpected failures arise.

Fix for upgrades already applied

To reconfigure the views and replace the references to NM3CONTEXT that may have escaped during the initial upgrade, unzip the files from fix release zip file into a staging folder. Copy the files into the folders as shown in the list of amended files above. This completes your install folders and ensures they are consistent with the product.

Then, log onto SQL*PLUS as the Highways Owner with the staging folder as the working directory.

At the prompt type "START nm_4500_fix20.sql" and press return.

Exit SQL*PLUS.

This script will perform the following – this is a cut-down version of the new upgrade process, executed in the correct order with potential for buffer-overflow removed.

- Install a new version of the nm3sdm package which will be used to generate spatial views
- Install new copies of the view definitions that drive the loop of spatial themes being addressed
- Remove extraneous subordinate user views, synonyms etc. if any remain after original upgrade.
- Use the modified code and views with the standard APIs to regenerate all spatial themes views.
- Refresh any view with a first-level dependency on NM3CONTEXT.GET_EFFECTIVE_DATE after replacing that reference with one made to a context variable.
- Refresh synonyms.
- Compile-all
- Log the upgrade

At the close, inspect the compile-all script output log and ensure that all objects are in a compiled state.

Issues

Internal Task ID		Support Log(s)
0112103	<p>The core upgrade script from 4.4.0.0 to 4.5.0.0 performs various changes designed to improve the security method and to improve performance. In particular, the script performs several operations to remove subordinate users private views - a method was used in previous releases to provide access to spatial data as some products failed to recognise some synonyms. The script to perform this cation could fail prematurely when the output buffer overflowed. This would occur in cases where a large number of subordinate users were being addressed. It's failure would result in some private views remaining when they should have been removed.</p> <p>The upgrade is also designed to remove the dependency on a PL/SQL function NM3CONTEXT.GET_EFFECTIVE_DATE from view definitions and replace it with a reference to SYS_CONTEXT. A script in the install folder called REGEN.SQL performed this operation by first refreshing the core and nsg spatial views using some product APIs. It then looped over remaining view definitions which had a dependency on the NM3CONTEXT package.</p> <p>Several problems have been detected in the combination of the scripts and within the body of each which until recently have not come to light until a minor issue with locator failing to zoom to a specific theme was raised as a help ticket.</p> <p>The issues and the repair are:</p> <ul style="list-style-type: none"> • The script to remove the subordinate private view definitions is executed in the body of the nm4400_nm4500_metadata_upg.sql which is after the script to refresh the spatial views. In cases where the system is configured to use private synonyms (which is rare), the upgrade to the product views will fail. The exception handling is such that the issue is reported but other views will fail to be refreshed. The failure arises at the point where the script attempts to create a private synonym for the new views when a private view already exists. The order of execution will be changed to prevent this. Hence a new script will be executed from within the body of the main upgrade before the execution of REGEN.SQL • The script to remove the subordinate private view definitions sets the server output on and uses DBMS_OUTPUT within the loop. This can fail due to the output buffer being blown. This is also corrected. • The core spatial upgrade has been modified to exclude the NSG views (which are refreshed later anyway). This is a minor improvement and poses no real problem. • The REGEN.SQL file has been modified to operate on views with a direct dependency on the NM3CONTEXT.GET_EFFECTIVE_DATE function. Whereas in the previous issue the script would attempt to modify all invalid views which had a dependency through the Oracle dictionary view USER_DEPENDENCIES. This dictionary view can include second level dependencies and in such cases, the view definition would have been refreshed using only the text from the USER_VIEWS dictionary. Under these circumstances, views defined without column alias' within the body of the select statement would be corrupted. Also, view definitions that include instead-of triggers were also included in the process. This could leave view definitions devoid of their instead-of triggers. This has been modified to use the full view definition, to address only views with a direct dependency where no instead-of triggers have been defined. Again, output sent via the DBMS_OUTPUT procedure can fail to address the full list of views due to the output buffer being blown. This has also been corrected. 	8001418343