Network Manager
Fix Release Notes

4.7.0.x Fix 17

Table of Contents

1. Introduction

This document defines the changes made to the Network Manager product for 4.7.0.x Fix 17 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 Bentley account manager.

2. Fix Details

Fix Details Baseline Release	4.7.0.x
Fix Description	Build Java Tools to extract and upload Shapefiles.
Prerequisites	
Implementation Instructions	The staging folder is the location of the folder that **exnm04070002en_updt17.zip** was extracted to (the folder containing this readme). **1. Installing Java Tools –** Please follow the instructions in **Section 4.1** of this document. **2. Running database scripts –** Log onto **SQL*PLUS** as the **Highways Owner** with the staging folder as the working directory. At the prompt type **START nm_4700_fix17.sql** and press return. **Exit** SQL*Plus
Limitations	Please check **Section 6** of this document.
Configuration Information	None
How To Test	Recommend full regression test
Rollback Strategy	Initially implement on a test environment

3. List of Amended Files

Filename	*Version*
gt-api-13-beta.jar	1.0
gt-data-13-beta.jar	1.0
gt-epsg-hsql-13-beta.jar	1.0
gt-jdbc-oracle-13-beta.jar	1.0
gt-main-13-beta.jar	1.0
gt-metadata-13-beta.jar	1.0
gt-opengis-13-beta.jar	1.0
gt-referencing-13-beta.jar	1.0
gt-shapefile-13-beta.jar	1.1
hsqldb-2.2.8.jar	1.0
jai_core-1.1.3.jar	1.0
jsr-275-1.0-beta-2.jar	1.0
jts-1.13.jar	1.0
ojdbc6.jar	1.0
sde2shp.jar	1.1
sdoapi.jar	1.0
sdoutl.jar	1.0
shp2sde.jar	1.1
vecmath-1.3.2.jar	1.0
xdb.jar	1.0
runcommand.fnw	1.0
CMDUtilities.java	1.0
mci_ldjava_11g.bat	1.0

4. Java Shapefile Tools

Java versions of **SHP2SDE** and **SDE2SHP** have been developed to allow for the extract and upload of Shapefiles, rather than using the ESRI supplied modules. These Java Tools are JAR files that are developed using APIs from *GeoTools (13-beta)* and *Oracle Spatial (11.2.0.2)*.

4.1 Installation Instructions

4.1.1 System Requirements –

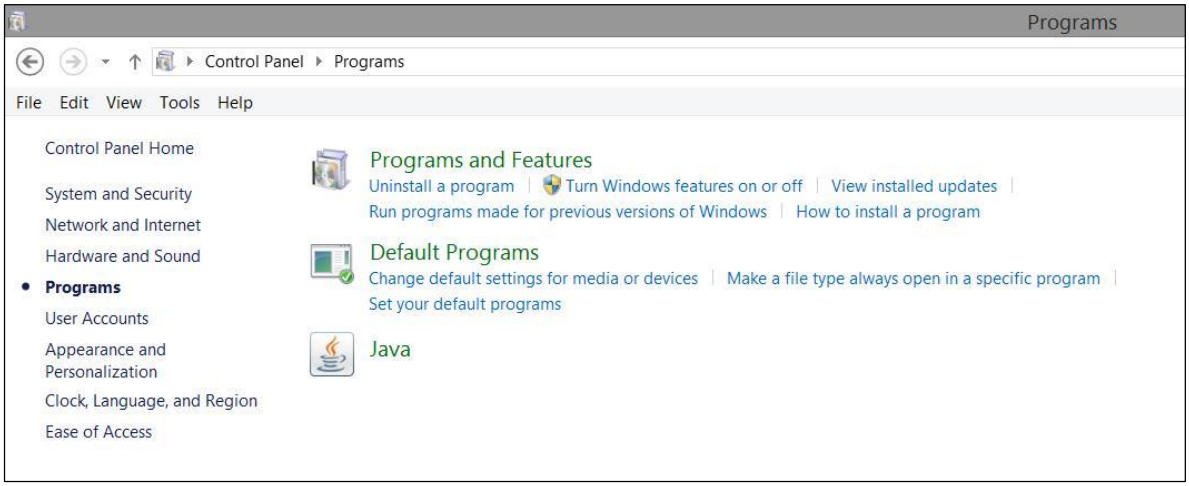
a. **Java JRE** – *JRE 1.7.0_25 64 bit* or *later* must be installed on the system from which the Java Shapefile Tools are to be executed.
b. **Database Server Access** – The system running Java Shapefile Tools must have access to the database server against which extract and upload operations are to be performed.

Note – In case the *Java Shapefile Tools* are to be used by the *PL/SQL way* (as explained in Section 4.2), installation steps given in Section 4.1.2 must be performed on the Database Server itself, as in this case, Database Server is the one from where the actual `java -jar` command will be executed. Also, the Shapefiles to be extracted and/or uploaded will reside on the Database Server itself.

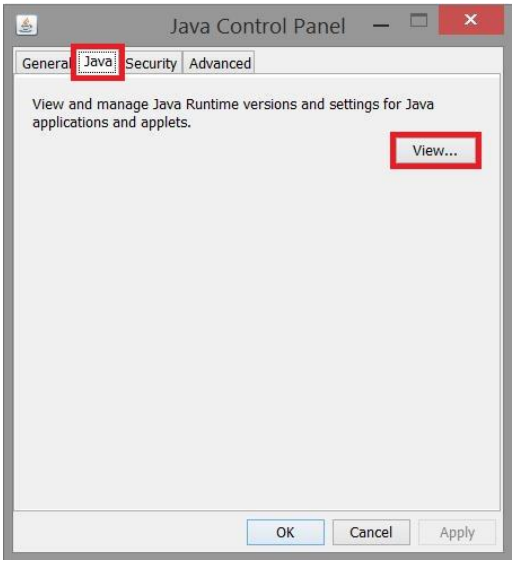
4.1.2 Steps to Install –

a. **Installing of Java Run Time** –
Install Java **JRE 1.7.0_25 64 bit** on the system. While installing make sure that the installation directory **does not contain any white spaces**.
e.g.
do not use – `C:\Program Files\Java\jre1.7.0_25`
use – `C:\Java\64\jre1.7.0_25`

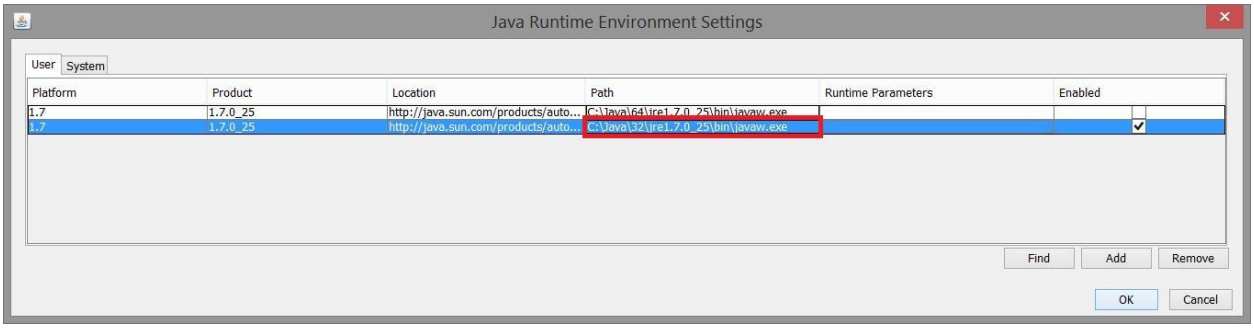
Open **Java Control Panel**: Control Panel -> Programs -> Java



Navigate to “Java” tab and click on “View…” button



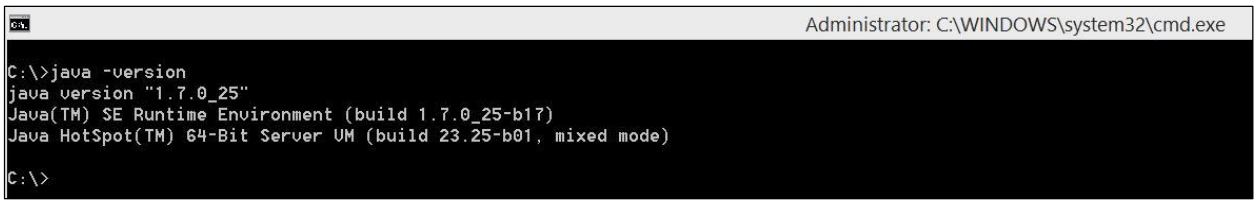
Check if the JRE version 1.7.0_25 64 bit is listed in the table by looking at the “Path” column.
If the table lists any other JREs uncheck the “Enabled” check box for them and check the same for 1.7.0_25 64 bit.



Open a **Command Window** and run the following command –

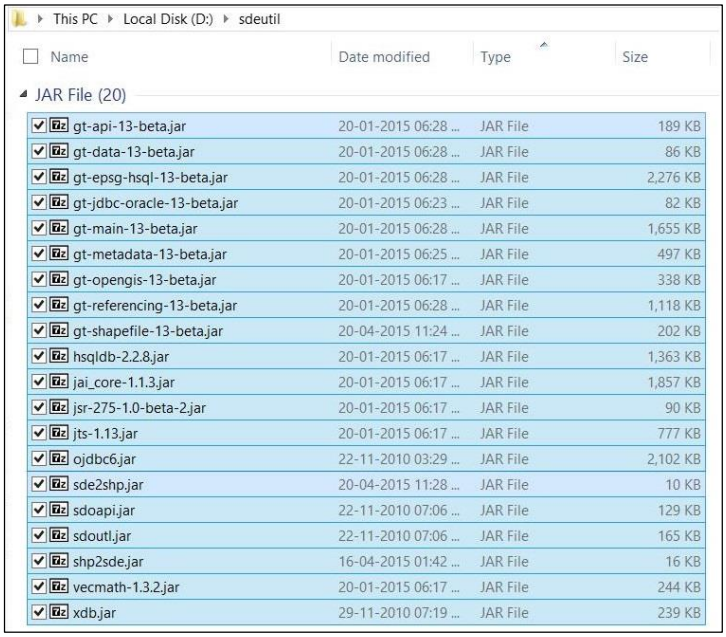
```
java -version
```

It should give an output similar to the following image –



b. **Copying the JAR files** –

Create a directory on the system with name **sdeutil** at a suitable location (again not containing any white spaces). Copy all the **jars** under **lib** folder from the staging folder to this directory.



c. **Creation of Additional Database Objects** –

Run **mci_ldjava_11g.bat** from **Command Prompt** with staging folder as working directory.

Usage: `mci_ldjava_11g.bat user/pass@connect`

Note –

1. The system executing this batch file must have **loadjava** utility. This can be confirmed by running the command – `loadjava -help` – on a **Command Prompt** on the system, if it shows the help options, the batch file can be run. Generally, **loadjava** is available on systems with Oracle Database or Oracle Database Client or Forms & Reports Server installed.
2. In case the system has *more than one Oracle Homes*, e.g. installing Oracle Database Server, Oracle Database Client – 32 bit and Oracle Database Client – 64 bit on the same system creates three Oracle Homes, PATH Environment Variable needs to be set before running the batch file on the **same Command Window**. **First entry** in the PATH variable must be the Oracle Home directory path for the **Database Server**.

3. `user` must be the *highways owner.*

4.2 Usage

Certain parameters need to be passed to the SHP2SDE and SDE2SHP jars. Following tables list all the Mandatory and Optional parameters and their descriptions.

a. **SHP2SDE** –

Parameter	Description
Mandatory Parameters	
-h	Host machine name/IP with existing Oracle database
-p	Host machine's port with existing Oracle database (e.g. 1521)
-s	Host machine's SID with existing Oracle database
-u	Database user's username
-d	Database user's password
-t	Table name for the result
-f	File name of an input Shapefile **without extension**
Optional Parameters	
-i	Column name for unique numeric ID; if required
-r	Valid Oracle SRID for coordinate system; use 0 if unknown
-g	Preferred or valid **SDO_GEOMETRY** column name
-x	Bounds for the X dimension; use -180,180 if unknown
-y	Bounds for the Y dimension; use -90,90 if unknown
-m	Load tolerance fields (x and y) in metadata, if not specified, tolerance fields are 0.05
-o	Mode to add Shapefile data to a table. Possible Values - {**append**|**create**|**init**} (values are **Case Sensitive**)
-n	Start ID for column specified in **-i** parameter
-c	Commit interval. Default, only commits at the end of a run.
-a	Attribute Mapping File **with extension**

Note – In case of *no projection* available for the Shapefile being uploaded (i.e. no `.prj` file), the SRID must be specified using `-r` command line arguments mentioned above. If not, the upload process will be terminated.

The priority to get the SRID will be –

1. `.prj` File
2. SRID specified with `-r` option

In both the above cases, if SRID differs the default SRID of the system to which the Shapefile is being uploaded, the upload process will be terminated.

b. **SDE2SHP** –

Parameter	Description
Mandatory Parameters	
-h	Host machine name/IP with existing Oracle database
-p	Host machine's port with existing Oracle database (e.g. 1521)
-s	Host machine's SID with existing Oracle database
-u	Database user's username
-d	Database user's password
-t	Input feature table name and spatial column name (separated by **comma** only)
-f	File name of an output Shapefile **without extension**
Optional Parameters	
-w	**WHERE** clause for the query
-a	Attribute Mapping File **with extension**

There are two ways to use these jars –

a. **Direct way** –

These jars can be directly executed through **Command Prompt** on the system, like –

```
java -jar "D:\sdeutil\sde2shp.jar" -h db_host -p db_port -s db_sid -u
db_username -d db_password -t db_tablename,column_name -w where_clause
-f shapefile_name -a attribute_map_file
```

b. **PL/SQL way** –

This fix creates a PL/SQL function in the database – **runcommand**. This can be used to execute the jars from within PL/SQL code as explained below –

In this scenario there are two terms –

i. **Command** - which we actually execute. There could be two types –
   - ✓ Actual command like `java -jar`...
   - ✓ Batch file - in case of batch file the commands line output will not be available from within PL/SQL context.

ii. **Process** - background process that runs the command.

The *runcommand* function takes three parameters –

i. **p_command** –

The command you want to execute as mentioned above.

ii. **p_success_str** –
In case you pass the command directly and not the batch file, and if the command writes some output on the *Command Prompt* on a *SEPARATE LINE*, you can compare that output with a string to perform some checks like success or failure. This is **Case Sensitive**.

iii. **p_output_mode** –
You can pass either of two values - a) **string** b) **integer**. These are **not Case Sensitive**.

- ✓ **string** returns
  - **If p_success_str is NOT NULL** –
    The whole command line output that is returned by the COMMAND being executed. This output will be trimmed to maximum size of *VARCHAR2* i.e. *32676*.
  - **If p_success_str is NULL** –
    It returns either *null* (in case PROCESS executed successfully) or *error* if PROCESS encounters some error.
- ✓ **integer** returns
  - **If p_success_str is NOT NULL** –
    1 for success and *-1* for failure of COMMAND.
  - **If p_success_str is NULL** –
    0/1 for success and *-1* for failure of PROCESS.

e.g.

```
SET SERVEROUTPUT ON
--
DECLARE
   v_output  INTEGER;
   v_command VARCHAR2(4000);
BEGIN
   v_command := 'java -jar "D:\sdeutil\sde2shp.jar" -h db_host -p db_port
   -s db_sid -u db_username -d db_password -t db_tablename,column_name -w
   where_clause -f shapefile_name -a attribute_map_file';
    --
   v_output := runcommand(v_command, 'success', 'integer');
   --
   dbms_output.put_line(v_output);
END;
/
```

Note – Any directories used in above commands must be already present on the system.

4.3 Logging

There are three logging levels depending on when and where error encounters –

4.3.1 **Shapefile Log File** –

It refers to a log file in the folder where the Shapefile being extracted or uploaded is kept and having same name as the Shapefile under operation. It contains information about various steps being followed while executing the extract or upload of the Shapefile and any errors encountered while executing the command.

This logging level generally logs errors related to wrong parameter-values and errors at database level like wrong username/password, table does not exist, shape column not found etc.

4.3.2 **System Log File** –

It refers to a log file contained in a folder – *log* – under the **sdeutil** directory where the Java Shapefile Tools jars are placed (refer Section 4.1.2 - b). This folder is created when extract/upload process is run for the first time. This log file contains a list of parameters and their values passed to the command (mentioned the tables of Section 4.2).

This logging level generally logs errors related to passing wrong number of parameters, absence of directories mentioned in the command and those that are thrown before creation of Shapefile Log file.

4.3.3 **Command Level Log** –

It refers to the command line output while executing the extract/upload process. In case of successful execution, the output shows –

i. Complete path to the System Log file.
ii. A message – "success" – indicating the process completed successfully. This message can be used in the *PL/SQL way* as explained in Section 4.2.

This logging level generally logs errors related to abnormal termination of the extract/upload process and those that are thrown before creation of System Log file.

5. Log No. Summary

This chapter summarises all software issues that have been addressed by this fix.

For issues raised by users, Bentley Technical Support Group (TSG) Service Request Numbers are cross referenced where applicable.

Details	*Internal Reference*	*TSG Service Request*
[NET] Build Java Tools to replace ESRI SHP2SDE and SDE2SHP Tools to upload and extract Shapefiles	Enhancement 205592	–
[MCI] Empty date attributes exports as "00000000" values	Defect 246542	–
[MCI] Too much decimals places at NM_START, NM_END attributes values	Defect 246564	–
[MCI] MCI fail uploading "no shape" asset file	Defect 246599	–

6. Known Issues

6.1 Column with data type NUMBER

Below are the restrictions depending on the format used for NUMBER data type –

1. **`NUMBER` –**
   It will be considered as `NUMBER (19, 9)`
2. **`NUMBER (precision)` –**
   In case precision is mentioned, minimum of `precision` or `19` will be used.
   e.g. DB Table - `NUMBER (10)` => Shapefile - `NUMBER (10)`
   DB Table - `NUMBER (25)` => Shapefile - `NUMBER (19)`
3. **`NUMBER (precision, scale)` –**
   GeoTools (13-beta) allows to set the size/precision for data type `NUMBER`, however not the scale. A proper solution on GeoTools side needs to be designed to allow this functionality.

   For now any NUMBER data type with scale specified will be considered as `NUMBER (19, 9)`

Note – Though the data type NUMBER (19, 9) means 10 digits before and 9 digits after the decimal point in terms of Oracle Standards (meaning total 19 digits excluding the decimal point), GeoTools considers the decimal point as a part of the total 19 characters. So, the value in such a column will be truncated/rounded off (the behaviour is unexpected) to one digit less –

e.g. 1234567890.123456789 will be stored in Shapefile to something like 1234567890.12345678

6.2 Column Name Length –

In a Shapefile, field/column names can be maximum of 10 characters long. If the database table/view has column names longer than 10 characters, those will be truncated to 10 characters in the corresponding DBF file. To avoid this use of *Attribute Mapping File* is recommended.

7. Changes in GeoTools Source Code

GeoTools being an open source API, below mentioned Java class files were edited to suit the requirements –

7.1 Jar – `gt-shapefile-13-beta.jar`
Class – `org.geotools.data.shapefile.ShapefileDataStore`

`ShapefileDataStore` – was forcing the data type `NUMBER` to be `NUMBER (33, 31)`. This was causing problem as it allows only two digits before decimal point and hence truncating numbers in essential records of database.

It was changed to make it `NUMBER (19, 9)` (See Section 6.1).

Precision and scale were selected to be 19 and 9 respectively to match the ESRI Shapefile specifications.

7.2 Jar – `gt-shapefile-13-beta.jar`
Class – `org.geotools.data.shapefile.dbf.DbaseFileWriter`

`DbaseFileWriter` – was replacing a NULL Date with eight '0's – default length for a Date type column in a Shapefile being eight characters.

It was changed to show null dates as `NULL` only instead of '0's.