



Network Manager Fix Release Notes

4.7.0.x Fix 17

Table of Contents

1. Introduction	2
2. Fix Details	2
3. List of Amended Files	3
4. Java Shapefile Tools	4
4.1 Installation Instructions	4
4.1.1 System Requirements	4
4.1.2 Steps to Install.....	4
4.2 Usgae	7
4.3 Logging	10
4.3.1 Shapefile Log File	10
4.3.2 System Log File	10
4.3.3 Command Level Log	10
5. Log No. Summary.....	11
6. Known Issues	11

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.



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	<p>The staging folder is the location of the folder that exnm04070001en_updt17.zip was extracted to (the folder containing this readme).</p> <ol style="list-style-type: none">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

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

3. List of Amended Files

<i>Filename</i>	<i>Version</i>
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	1.0
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	1.0
sdoapi.jar	1.0
sdoutl.jar	1.0
shp2sde.jar	1.0
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

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.



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* 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 –**

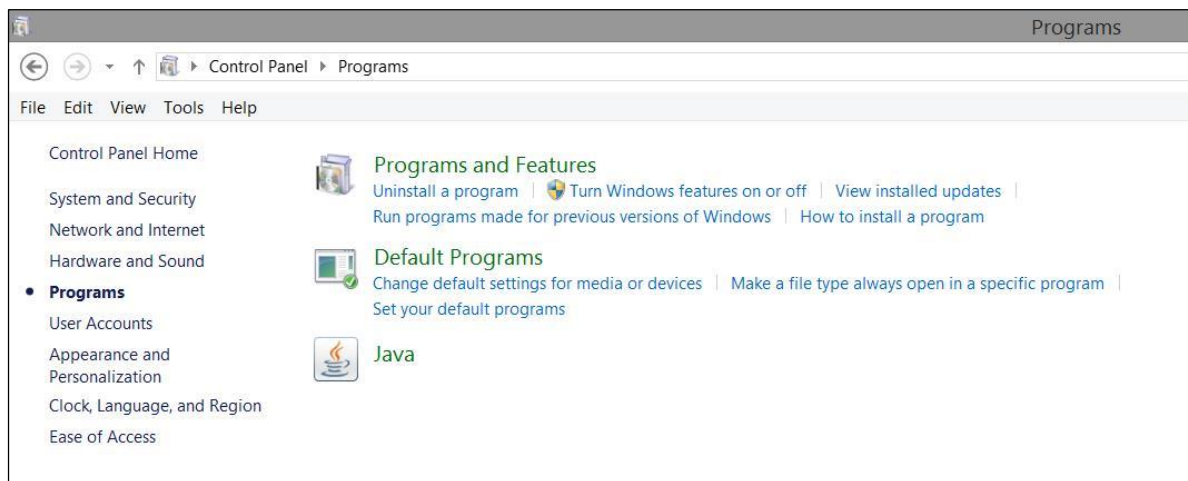
Install **Java JRE 1.7.0_25 64 bit** on the system. While installing make sure that the installation directory **does not contain any white spaces**.

e.g.

do not use – C:\Program Files\Java\jre1.7.0_25

use – C:\Java\64\jre1.7.0_25

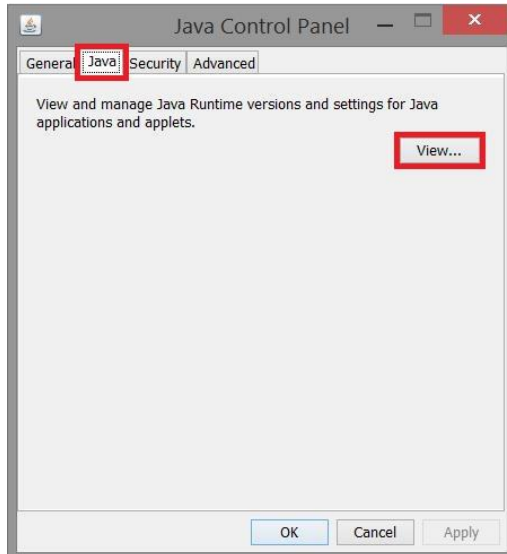
Open **Java Control Panel**: Control Panel -> Programs -> Java



CONFIDENTIALITY STATEMENT

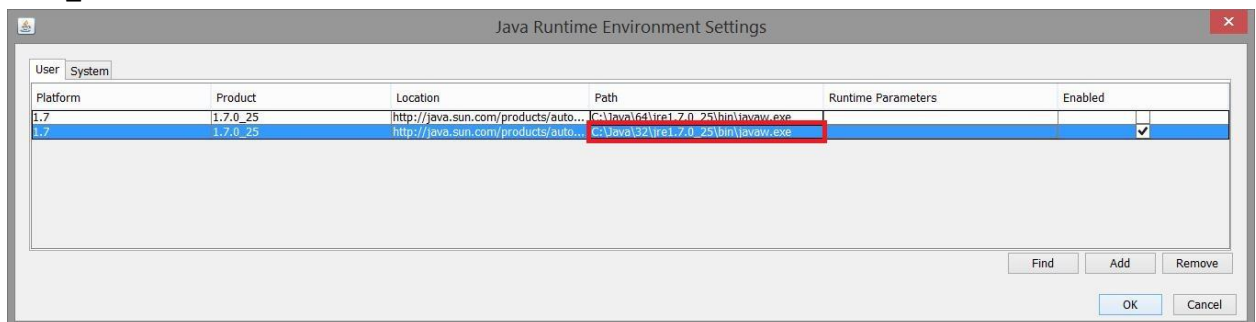
The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

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.

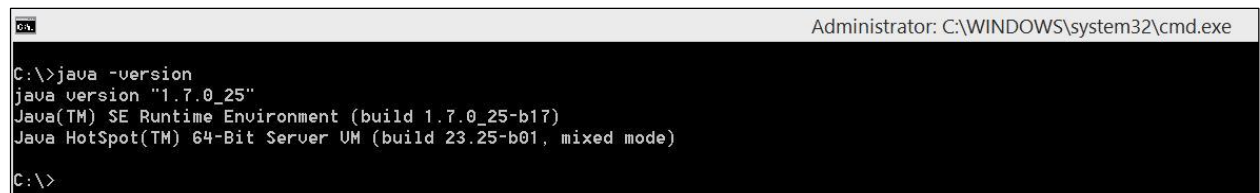
If the table lists any other JREs uncheck the “Enabled” check box for them and check the same for 1.7.0_25 64 bit.



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

```
java -version
```

It should give an output similar to the following image –



```
C:\>java -version
java version "1.7.0_25"
Java(TM) SE Runtime Environment (build 1.7.0_25-b17)
Java HotSpot(TM) 64-Bit Server VM (build 23.25-b01, mixed mode)
C:\>
```

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

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.



Name	Date modified	Type	Size
gt-api-13-beta.jar	20-01-2015 06:28 ...	JAR File	189 KB
gt-data-13-beta.jar	20-01-2015 06:28 ...	JAR File	86 KB
gt-epsg-hsql-13-beta.jar	20-01-2015 06:28 ...	JAR File	2,276 KB
gt-jdbc-oracle-13-beta.jar	20-01-2015 06:23 ...	JAR File	82 KB
gt-main-13-beta.jar	20-01-2015 06:28 ...	JAR File	1,655 KB
gt-metadata-13-beta.jar	20-01-2015 06:25 ...	JAR File	497 KB
gt-opengis-13-beta.jar	20-01-2015 06:17 ...	JAR File	338 KB
gt-referencing-13-beta.jar	20-01-2015 06:28 ...	JAR File	1,118 KB
gt-shapefile-13-beta.jar	20-04-2015 11:24 ...	JAR File	202 KB
hsqldb-2.2.8.jar	20-01-2015 06:17 ...	JAR File	1,363 KB
jai_core-1.1.3.jar	20-01-2015 06:17 ...	JAR File	1,857 KB
jsr-275-1.0-beta-2.jar	20-01-2015 06:17 ...	JAR File	90 KB
jts-1.13.jar	20-01-2015 06:17 ...	JAR File	777 KB
ojdbc6.jar	22-11-2010 03:29 ...	JAR File	2,102 KB
sde2shp.jar	20-04-2015 11:28 ...	JAR File	10 KB
sdoapi.jar	22-11-2010 07:06 ...	JAR File	129 KB
sdoutl.jar	22-11-2010 07:06 ...	JAR File	165 KB
shp2sde.jar	16-04-2015 01:42 ...	JAR File	16 KB
vecmath-1.3.2.jar	20-01-2015 06:17 ...	JAR File	244 KB
xdb.jar	29-11-2010 07:19 ...	JAR File	239 KB

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**.

```

C:\Stage\exnm04070001en_updt17>SET PATH=D:\Oracle\Product\11.2.0.2\Bentley;%PATH%
C:\Stage\exnm04070001en_updt17>mci_ldjava_11g.bat user/password@connect

```

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

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.



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

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

**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.

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

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.

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.

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	–

6. Known Issues

6.1 Size for data type NUMBER

A class from Java API – `java.math.BigDecimal` – represents the `NUMBER` data type in Oracle Database. While extracting a Shapefile, the Oracle Database data types are mapped to corresponding Java API classes. While mapping for `NUMBER` data type, one of the classes from GeoTools (13-beta) API – `org.geotools.data.shapefile.ShapefileDataStore` – was forcing it to the data type `NUMBER (33, 31)`. This was causing problem as this data type allows only two digits before decimal point and hence truncating numbers in essential records from database, hence loss of precision. Also `NUMBER` data type can store a number with maximum 38 digits, hence maximum potential was not used.

GeoTools being an open source API, the above mentioned Java class file was edited to force the data type to be – `NUMBER (38, 19)`. It allows 19 digits both before and after the decimal point which are sufficient for now as per Exor's current requirements.

GeoTools (13-beta) allows to set the size for a data type including `NUMBER`, however not to set the scale (number of digits after decimal) for `NUMBER` data type. A proper solution on GeoTools side needs to be designed to allow this functionality.

CONFIDENTIALITY STATEMENT

The contents of this document, including system ideas and concepts, are confidential and proprietary in nature and are not to be distributed in any form without the prior written consent of Bentley Systems, Inc.