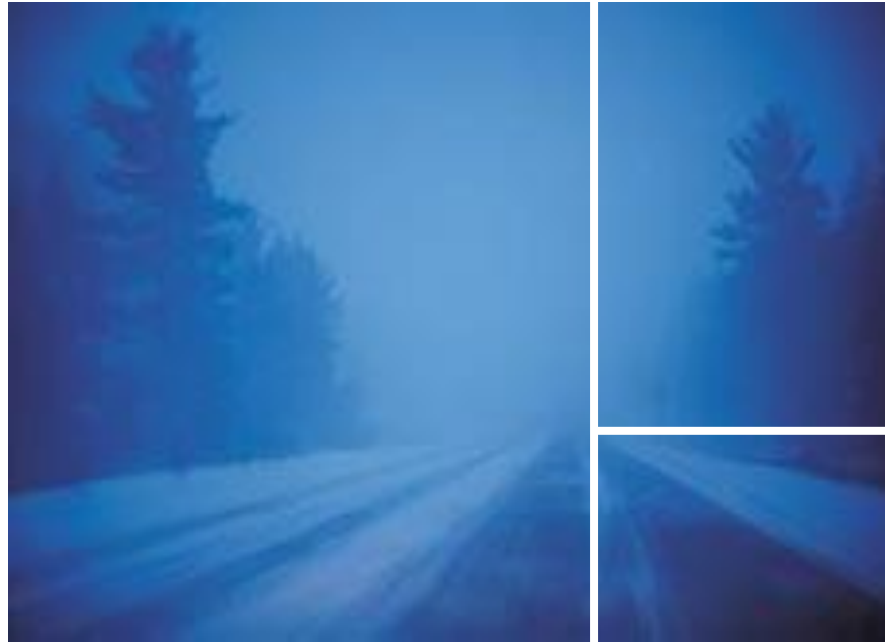




F

Exor Corporation Limited



CSV Data Loader v4.3



***The Global Leader in
Infrastructure Asset Management***



This page intentionally blank

Quality Assurance Statement

Document Details	
File: CSV Data Loader v4.3.pdf	Prepared by: T.C. Stewart
Manual Name: CSV Data Loader Guide	Reviewed by:
Version: 3.0	Approved for issue by:
Date of Issue: November 2010	Product Manager:
File Name: CSV Data Loader v4.3.pdf	

Document Version Control				
Revision	Date	By	Page	Description
1.0	18-Dec-06	CS	12	V_LOAD_LOCATE_INV_BY_REF
1.1	18-Dec-06	CS	29	Loading Asset Data using Ref Post Referencing
3.0	November 2010	IS		Updated for 4.3 release

CONTENTS

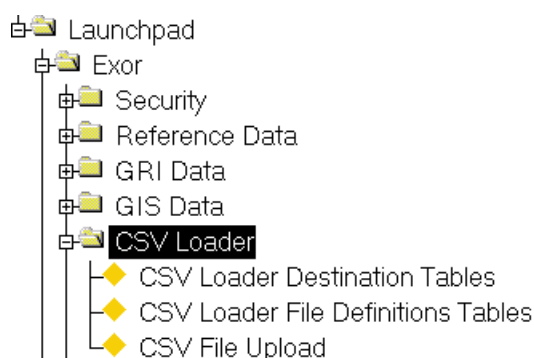
Quality Assurance Statement	2
CSV Loader Destination Table - HIG2010.....	5
Destination Tables	6
Default Values Panel	7
Loading Asset Item Data	8
Loading Asset Item Location Data	8
V_LOAD_INV_MEM_ON_ELEMENT.....	8
V_LOAD_INV_MEM_ELE_MP_EXCL	9
V_LOAD_INV_MEM_ELE_MP_AMBIG	11
V_LOAD_LOCATE_INV_BY_REF	12
Loading Network Node Points.....	14
Loading Network Elements (Datum or Groups)	15
CSV Loader File Definitions Table - HIG2020.....	16
File Definitions Table	17
File Columns Panel.....	18
Creating a Holding Table	19
File Destinations Panel	20
Loading Asset	22
File Column Settings.....	23
File Destination Settings	24
NM_INV_ITEMS (Asset Item Details).....	24
V_LOAD_INV_MEM_ON_ELEMENT (Asset Item Locations).....	26
Loading Asset Data using Reference Post Referencing	29
File Column Settings.....	31
File Destination Settings	32
NM_INV_ITEMS (Asset Item Details).....	32
V_LOAD_LOCATE_INV_BY_REF (Locate Assets using Reference Post Referencing).....	34
Loading Network Nodes.....	36
File Column Settings.....	37
File Destination Settings	38
NM_POINTS.....	38
NM_NODES_ALL	39
Loading Network Elements (Datum or Groups)	40
File Columns.....	41
File Destinations	41
Loading Group Members	47
File Column Settings.....	48
File Destinations	48
Load/Process Data - HIGWEB2030	55
CSV File Upload Process.....	55
Logging onto a Web Session	57
Hyper Links.....	58
Selecting the File Definition.....	58
Load New Batch from Server	60
Load New Batch from Client	62
Load / Validate Data (2nd Stage)	65
Batch Housekeeping.....	66

CHAPTER

1

Introduction

Figure 1
CSV Loader Menu



The CSV Data Loader is a Generic loader, which allows data to be loaded into any of the Tables within **Exor**. For example it could be used to load Asset Item Data, Network Data, Document Associations within **Document Manager**, or Asset Domains and Lookup to name but a few.

The Load file may reside either on the Client Machine or on the Database Server Machine.

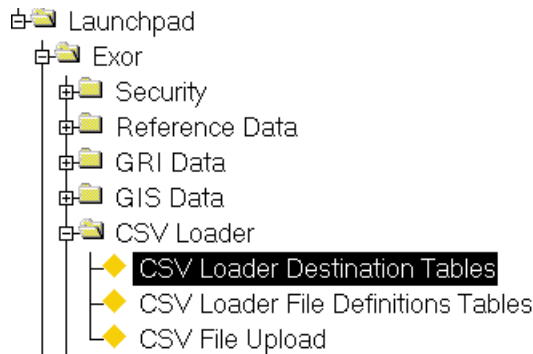
The loading of data using the CSV Loader is essentially a 2-stage process. Data is firstly loaded into a temporary or 'Holding' table then transferred to the appropriate Table within **Exor**. Data validation is carried out during both stages as described in Chapter 2 of this document.

This Chapter describes the modules used to define the required Destination Tables/ Insert and Validation procedure combinations and those required to define the structure of the Holding Table and the Mapping of the Data between the Holding and Destination Tables. The following modules are included:

- CSV Loader Destination Tables - HIG2010
- CSV Loader File Definitions Table - HIG2020

CSV Loader Destination Table - HIG2010

Figure 2 CSV Loader Destination tables Menu Option

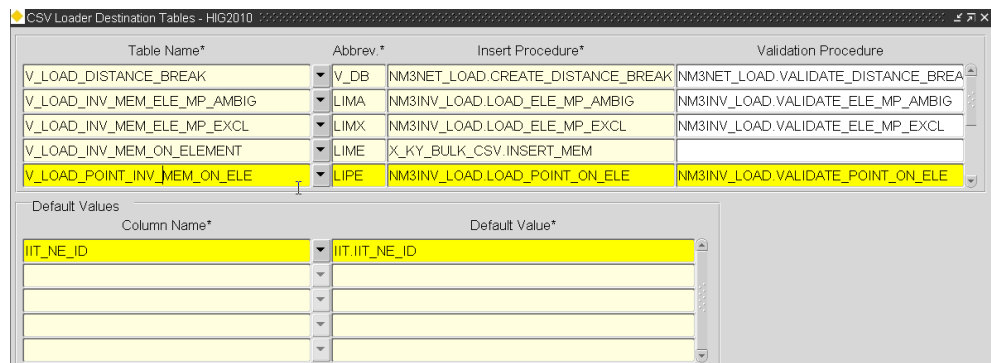


The **CSV Loader Destination Table - HIG2010** module is used to define the Destination Table(s) (this may be a Database View) that data will be loaded into during the '2nd stage' of the Loading process. The Insert and Validation Procedures used during the transfer of data from a holding table to the specified Destination table(s) are also defined within this module.

Standard metadata is supplied and installed during new installations or upgrades of Exor, which will define the required Destination Tables, Insert and Validation procedures required for loading data such as Asset Items, Network Nodes and Elements.

Columns within a Destination Table must be 'mapped' to the appropriate Column in 'Holding' or temporary table. This is done using the **CSV Loader Files Definitions Tables - HIG2020** module. When mapping Columns within a Destination Table to the appropriate Column in a Holding table, default values may be used to autopopulate the 'Source Column'. This is useful when the source of the data is a call to a Database Procedure which returns a specific value. An example of this would be the Default source value of the **IIT_NE_ID** column when loading Asset Data. This is the internal Unique of the Asset Item and would not normally be contained within the 'raw' CSV Data file. Using the Procedural Call '**NM3NET.GET_NEXT_NE_ID**' in the Source column will ensure that the appropriate value is loaded for the field. Default values which are automatically populated in the Source Column of the 'File Destinations' panel of the **CSV Loader Files Definitions Tables - HIG2020** module may be overwritten or amended if required.

Figure 3



The screenshot shows the 'CSV Loader Destination Tables - HIG2010' configuration window. It contains a table with columns: Table Name*, Abbrev.*, Insert Procedure*, and Validation Procedure. Below this table is a section for 'Default Values' with columns for Column Name* and Default Value*.

Table Name*	Abbrev.*	Insert Procedure*	Validation Procedure
V_LOAD_DISTANCE_BREAK	V_DB	NM3NET_LOAD.CREATE_DISTANCE_BREAK	NM3NET_LOAD.VALIDATE_DISTANCE_BREA
V_LOAD_INV_MEM_ELE_MP_AMBIG	LIMA	NM3INV_LOAD.LOAD_ELE_MP_AMBIG	NM3INV_LOAD.VALIDATE_ELE_MP_AMBIG
V_LOAD_INV_MEM_ELE_MP_EXCL	LIMX	NM3INV_LOAD.LOAD_ELE_MP_EXCL	NM3INV_LOAD.VALIDATE_ELE_MP_EXCL
V_LOAD_INV_MEM_ON_ELEMENT	LIME	X_KY_BULK_CSV.INSERT_MEM	
V_LOAD_POINT_INV_MEM_ON_ELE	LIPE	NM3INV_LOAD.LOAD_POINT_ON_ELE	NM3INV_LOAD.VALIDATE_POINT_ON_ELE

Column Name*	Default Value*
IIT_NE_ID	IIT.IIT_NE_ID

Destination Tables

Table Name	(Required)	List
Enter the Table or View name into which Data will be loaded during the 'transfer' from a Holding Table. For example, when loading Asset Item details the value would be NM_INV_ITEMS.		

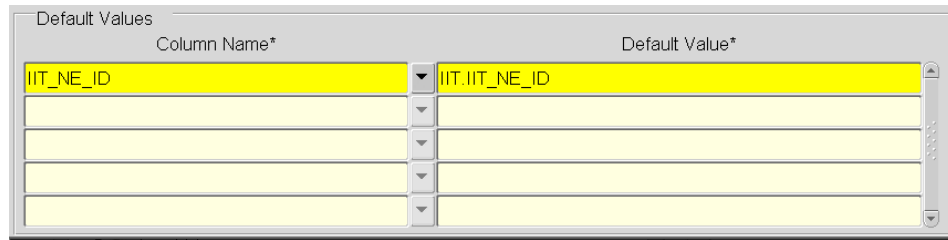
Note that in the case of loading Asset Data a different Table or View would be defined for the Asset Locations.

Abbrev	(Required)
Enter an Abbreviation for the Table/View name. This must be Unique not only across other Abbreviations used but also across the Unique Reference given to a File Definition Table in the CSV Loader Files Definitions Tables - HIG2020 module. This Abbreviation may be used as an alias when mapping a Destination Column to a Source Column where the Source column is contained within another Destination Table. A maximum of 5 characters is allowed.	

Insert Procedure	(Required)
Enter the name of the Insert procedure used when loading data from a Holding table into the specified Destination table. When loading Asset Data into NM_INV_ITEMS this value will be NM3INS.INS_IIT_ALL	

Validation Procedure
If required enter the name of a Validation procedure used to validate data during the loading of data from a holding table to the specified Destination table. When loading Asset Data into NM_INV_ITEMS this value will be NM3INV.VALIDATE_REC_IIT .

Figure 4



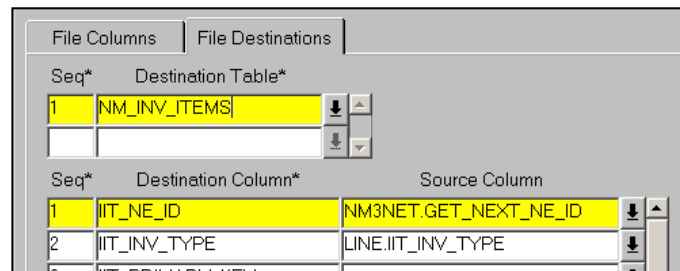
Column Name*	Default Value*
IIT_NE_ID	IIT.IIT_NE_ID

Default Values Panel

This panel is used to define Default values for the required Columns within the selected Destination Table. The Default value specified will be automatically populated in the 'Source Column' field of the File Destinations panel in the **CSV Loader Files Definitions Tables - HIG2020** module when mapping the Destination and Source Columns for the selected Table. The Default value defined may be of any datatype (character strings should be enclosed in single quotes and date should be wrapped inside a TO_DATE function, e.g. TO_DATE('01-JAN-2000','DD-MON-YYYY')) or may be a call to a database Procedure as in the example shown in Figure 4. Standard Oracle Formatting Functions may also be used, e.g. LTRIM(TO_CHAR(<fieldname>), ' ')

Figure 5 shows an extract of the **CSV Loader Files Definitions Tables - HIG2020** module where the Default Value for the IIT_NE_ID column in NM_INV_ITEMS has been defaulted to the value specified for the Column in the **CSV Loader Destination Table - HIG2010** as shown in Figure 4.

Figure 5



Seq*	Destination Table*	Source Column
1	NM_INV_ITEMS	NM3NET.GET_NEXT_NE_ID
2	IIT_INV_TYPE	LINE.IIT_INV_TYPE
3	IIT_PRIMARY_KEY	

Column Name

List

Enter the Column name for which to define a Default Value.

Default Value

Enter the required default Value.

Loading Asset Item Data

The Destination Table table (and associated Insert/Validation procedures) used when loading Asset Item details is the **NM_INV_ITEMS** table.

Table 1 shows the required values for the CSV Loader Destination Tables in order to load Asset Item data. **This data is supplied as standard metadata.**

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 1

Table Name	Abbreviation	Insert Procedure	Validation Procedure
NM_INV_ITEMS	IIT	NM3INS.INS_IIT_A LL	NM3INV.VALIDATE_REC_IIT

Table 2 shows the default values used as the 'Data Source' for the NM_INV_ITEMS Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the NM_INV_ITEMS Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 2

Column Name	Default Value
IIT_NE_ID	nm3net.get_next_ne_id

Loading Asset Item Location Data

The Destination Table and Insert/Validation procedure combination used when loading Asset data locations is dependant upon the Network Model and Route configuration defined for the network on which the data is located.

The following Destination Tables (along with the appropriate Insert and Validation procedures) are supplied:

- V_LOAD_INV_MEM_ON_ELEMENT
- V_LOAD_INV_MEM_ELE_MP_EXCL
- V_LOAD_INV_MEM_ELE_MP_AMBIG
- V_LOAD_LOCATE_INV_BY_REF

V_LOAD_INV_MEM_ON_ELEMENT

This Destination Table and Insert/Validation procedure combination should be used if the location of the Asset Data is to be loaded by Route and Offset and where Sub Class has **NOT** been used as a Carriageway identifier on the Datum Elements, i.e. a Route Offset is never ambiguous. If Route Offsets may be ambiguous the **V_LOAD_INV_MEM_ELE_MP_EXCL** or **V_LOAD_INV_MEM_ELE_MP_AMBIG** Destination Tables should be used.

Table 3 shows the required values for the CSV Loader Destination Tables in order to load Asset Item Locations using a non ambiguous Route and Offset. This data is supplied as standard metadata.

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 3

Table Name	V_LOAD_INV_MEM_ON_ELEMENT
Abbrev	LIME
Insert Procedure	NM3INV_LOAD.LOAD_ON_ELEMENT
Validation Procedure	NM3INV_LOAD.VALIDATE_ON_ELEMENT

Table 4 shows the default values used as the 'Data Source' for the V_LOAD_INV_MEM_ON_ELEMENT Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the Destination Table in module *CSV Loader File Definitions Table - HIG2020*.

Table 4

Column Name	Default Value
IIT_NE_ID	IIT.IIT_NE_ID

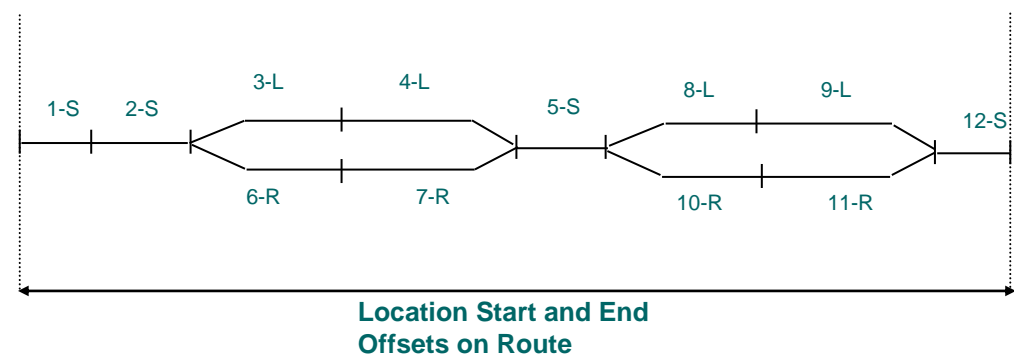
Note that the 'IIT' value which prefixes IIT_NE_ID in the Default Value field is the Abbreviation used for the NM_INV_ITEMS Destination table.

V_LOAD_INV_MEM_ELE_MP_EXCL

This Destination Table and Insert/Validation procedure combination should be used if the location of the Asset Data is to be loaded by Route and Offset where Sub Class has been used as a Carriageway identifier on the Datum Elements and the Asset Locations are to be restricted to the Network locations of a single 'Exclusive' Sub Class.

Consider the example in Figure 6.

Figure 6



The Route displayed in Figure 6 contains a combination of Network Type Subclasses. An item of Asset is to be located within the extent limits shown, but only where the Network Type Sub Class is 'L' – Left.

By using the V_LOAD_INV_MEM_ELE_MP_EXCL Destination Table and Insert/Validation procedure combination and providing the exclusive Sub

Class in the 'raw data file' (this maps to the EXCLUSIVE_SUBCLASS destination column when defining the column 'mappings' for the load file in the **CSV Loader File Definitions Table - HIG2020** module (page 4)) the location of the Asset may be restricted to a single Subclass.

If an exclusive Sub Class of '**L**' - **Left** was provided within the load file record , the Asset Item will only be located on Elements (or parts of) with a Network Type Subclass of 'L' – Left. Therefore the Asset Location would include the following Elements: 3-L, 4-L, 8-L, 9-L

Table 5 shows the required values for the CSV Loader Destination Tables in order to load Asset Item Locations by specifying an exclusive Network Type Sub Class, Route and Offset. This data is supplied as standard metadata.

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 5

Table Name	V_LOAD_INV_MEM_ELE_MP_EXCL
Abbrev	LIMX
Insert Procedure	NM3INV_LOAD.LOAD_ELE_MP_EXCL
Validation Procedure	NM3INV_LOAD.VALIDATE_ELE_MP_EXCL

Table 6 shows the default values used as the 'Data Source' for the V_LOAD_INV_MEM_ELE_MP_EXCL Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 6

Column Name	Default Value
IIT_NE_ID	IIT.IIT_NE_ID

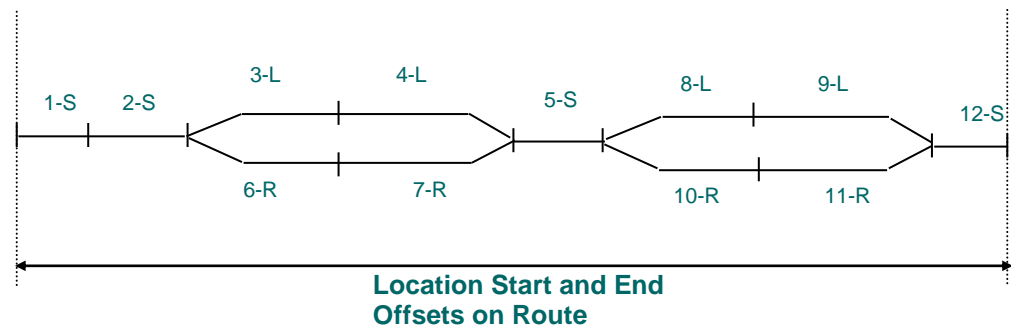
Note that the 'IIT' value which prefixes IIT_NE_ID in the Default Value field is the Abbreviation used for the NM_INV_ITEMS Destination table.

V_LOAD_INV_MEM_ELE_MP_AMBIG

This Destination Table and Insert/Validation procedure combination should be used if the location of the Asset Data is to be loaded by Route and Offset where Sub Class has been used as a Carriageway identifier on the Datum Elements and any Ambiguous location references are to be resolved by specifying the required Network Type Sub Class when an ambiguity is detected.

Consider the example in Figure 7

Figure 7
Extent limits



The Route displayed in Figure 7 contains a combination of Network Type Subclasses. An Ass Item is to be located within the extent limits shown, but only where the Network Type Sub Class is either 'S' – Single or 'L' – Left.

This can be done by using the V_LOAD_INV_MEM_ELE_MP_AMBIG Destination Table and Insert/Validation procedure combination and providing the required Sub Class to use when an ambiguity is detected, in the 'raw data file' (this maps to the SUBCLASS_WHEN_AMBIGUOUS destination column when defining the column 'mappings' for the load file in the **CSV Loader File Definitions Table - HIG2020** module (page 4)). If a value of 'L' – left was provided in the load file record, the Asset Item will only be located on Elements (or parts of) with a Network Type Subclass of 'S' – Single and those Elements (or parts of) with a Network Type Sub Class of 'L' –Left. Therefore the Asset Location would include the following Elements: 1-S, 2-S, 3-L, 4-L, 5-S, 8-L, 9-L, 12-S

Table 7 shows the required values for the CSV Loader Destination Tables in order to load Asset Item Locations using an ambiguous Route and Offset. This data is supplied as standard metadata.

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 7

Table Name	V_LOAD_INV_MEM_ELE_MP_AMBIG
Abbrev	LIMA
Insert Procedure	NM3INV_LOAD.LOAD_ELE_MP_AMBIG
Validation Procedure	NM3INV_LOAD.VALIDATE_ELE_MP_AMBIG

Table 8 shows the default values used as the 'Data Source' for the V_LOAD_INV_MEM_ELE_MP_AMBIG Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 8

Column Name	Default Value
IIT_NE_ID	IIT.IIT_NE_ID

Note that the 'IIT' value which prefixes IIT_NE_ID in the Default Value field is the Abbreviation used for the NM_INV_ITEMS Destination table.

V_LOAD_LOCATE_INV_BY_REF

This Destination Table and Insert procedure combination should be used if the location of the Asset Data is to be loaded using Route / Reference Post referencing (MilePost referencing).

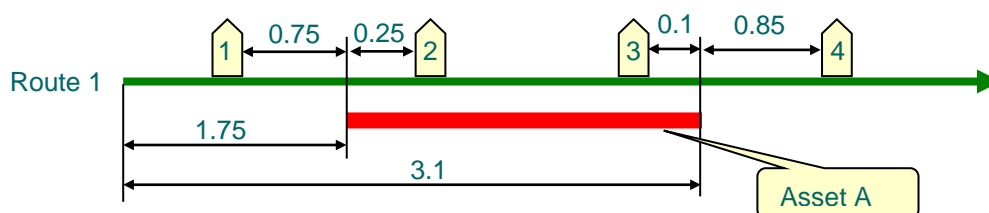
Reference Post or 'Route Mile Post' referencing, allows the location of an Asset to be expressed relative to the distance from a known 'reference' Asset, located on the selected Route. The 'Reference' Asset must be a 'Point Item' and can be either an Asset stored within the Exor database or an External Asset.

The 'Reference Post Offset' is always entered relative to the Route cardinality. Therefore a positive distance from a Reference Post means that the Asset is located further along the Route than the 'reference Post', where as a negative distance means the Asset is located on the Route before the 'reference post'.

If no reference Item is specified, the Reference Offset is taken as the distance from the Start of the Route.

Consider the example in Figure 8.

Figure 8



Asset 'A' could be located in several different ways relative to the Marker Posts (MP) shown in the diagram. Its start location could be expressed as follows.

Route	MP	Offset
1	1	0.75
Or		
Route	MP	Offset
1	2	-0.25
Or		
Route	MP	Offset
1	null	1.75

Its end location could be expressed as follows:

Route	MP	Offset
1	3	0.1
Or		
Route	MP	Offset
1	4	-0.85
Or		
Route	MP	Offset
1	null	3.1

Table 9 shows the required values for the CSV Loader Destination Tables in order to load Asset Item Locations using Reference Post Referencing. This data is supplied as standard metadata.

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 9

Table Name	V_LOAD_LOCATE_INV_BY_REF
Abbrev	LIBR
Insert Procedure	NM3MP_REF.LOCATE_ASSET

Table 10 shows the default values used as the 'Data Source' for the V_LOAD_LOCATE_INV_BY_REF Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the Destination Table in module *CSV Loader File Definitions Table - HIG2020*.

Table 10

Column Name	Default Value
IIT_NE_ID	IIT.IIT_NE_ID

Note that the 'IIT' value which prefixes IIT_NE_ID in the Default Value field is the Abbreviation used for the NM_INV_ITEMS Destination table.

Loading Network Node Points

When loading Network Nodes there are 2 Destination Tables which must be defined. These are namely the **NM_POINTS** and **NM_NODES_ALL** tables.

Table 11 shows the required values for the CSV Loader Destination Tables in order to load Network Nodes. **This data is supplied as standard metadata.**

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 11

Table Name	Abbrev	Insert Procedure
NM_NODES_ALL	NO	NM3INS.INS_NO_ALL
NM_POINTS	NP	NM3INS.INS_NP

Note that a Validation procedure is not normally required.

Table 12 shows the default value used as the 'Data Source' for the **NM_NODES_ALL** Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the **NM_NODES_ALL** Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 12

Column Name	Default Value
NO_NP_ID	np.np_id

Table 13 shows the default value used as the 'Data Source' for the **NM_POINTS** Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the **NM_POINTS** Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 13

Column Name	Default Value
NP_ID	nm3seq.next_np_id_seq

Loading Network Elements (Datum or Groups)

When loading Network Elements (Datum or Groups) the Destination Table used is NM_ELEMENTS_ALL with an insert procedure of NM3NET.INSERT_ANY_ELEMENT.

Table 14 shows the required values for the CSV Loader Destination Tables in order to load Network Elements. **This data is supplied as standard metadata.**

Note that the table abbreviation may be amended if required, but it is not normally necessary to do so.

Table 14

Table Name	Abbrev	Insert Procedure
NM_ELEMENTS_ALL	NE	NM3NET.INSERT_ANY_ELEMENT

Note that a Validation procedure is not normally required.

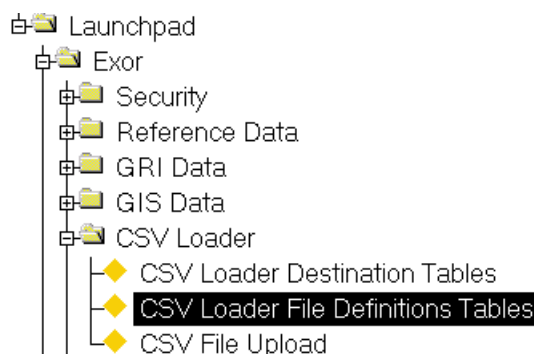
Table 15 shows the default value used as the 'Data Source' for the **NM_ELEMENTS_ALL** Destination Table. This will be seen when 'mapping' the Columns in the 'Holding' or temporary Table to the **NM_ELEMENTS_ALL** Destination Table in module **CSV Loader File Definitions Table - HIG2020**.

Table 15

Column Name	Default Value
NE_ID	nm3seq.next_NE_ID_SEQ

CSV Loader File Definitions Table - HIG2020

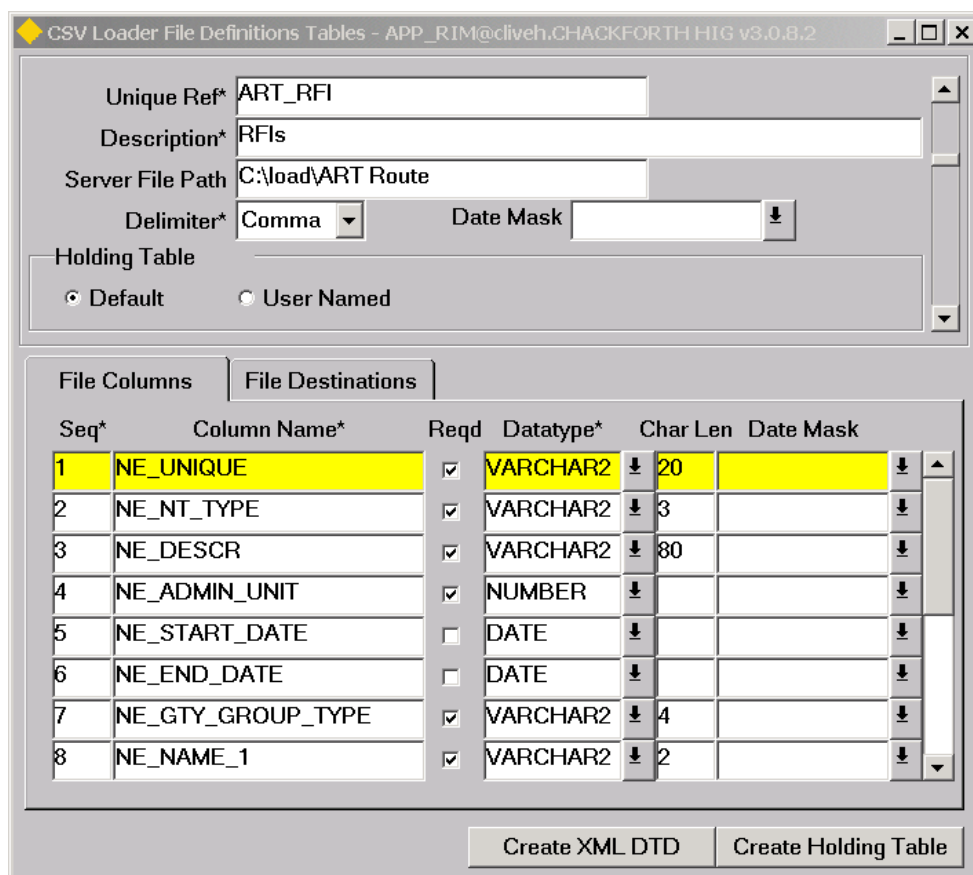
Figure 9
CSV Loader
File Def table
Menu Option



The *CSV Loader File Definitions Table - HIG2020* module is used to define and optionally create the Holding Table (this should match the 'Raw' data file layout) used in the first stage of the Loading process and to map the Columns in the appropriate Destination table with the data Source. The Source of the data will generally be from within the Holding Table, but may also be a Column within another Destination Table or a call to a database Procedure.

The module is also used to define the directory path, as seen by the Server, where the data file to be loaded resides and the Delimiter used within the file.

Figure 10



Unique Ref* ART_RFI

Description* RFIs

Server File Path C:\load\ART Route

Delimiter* Comma Date Mask

Holding Table

☒ Default ☐ User Named

Seq*	Column Name*	Reqd	Datatype*	Char Len	Date Mask
1	NE_UNIQUE	<input checked="" type="checkbox"/>	VARCHAR2	20	
2	NE_NT_TYPE	<input checked="" type="checkbox"/>	VARCHAR2	3	
3	NE_DESCR	<input checked="" type="checkbox"/>	VARCHAR2	80	
4	NE_ADMIN_UNIT	<input checked="" type="checkbox"/>	NUMBER		
5	NE_START_DATE	<input type="checkbox"/>	DATE		
6	NE_END_DATE	<input type="checkbox"/>	DATE		
7	NE_GTY_GROUP_TYPE	<input checked="" type="checkbox"/>	VARCHAR2	4	
8	NE_NAME_1	<input checked="" type="checkbox"/>	VARCHAR2	2	

Create XML DTD Create Holding Table

File Definitions Table

Unique Ref (Required)

Enter a Unique name for the File Definition Table. This must be Unique not only across other Definition Table Unique Refs but also across the Abbreviations given to Destination Tables in the **CSV Loader Destination Table - HIG2010** module. This Unique Ref will be used as an alias when mapping a Destination Column to a Source Column and will also be used in the naming convention of the Holding Table. A maximum of 20 characters is allowed.

Note that spaces are not permitted in the Unique Ref, underscore should be used in their place where required.

Description (Required)

Enter a description for the File Definition Table. A maximum of 80 characters is allowed.

Server File Path (Optional)

If the file is to be loaded is located on the Server machine, enter the directory path on the Server machine, where the data file is located. This specified directory path must be the directory or a sub directory at a lower level as defined within the utl_file_dir setting of the init.ora file.

The data file may also be loaded directly from the Client Machine. See page 62 for details of this operation.

Delimiter (Required) List

Select the type of Delimiter used within the dat file to be loaded. The list of allowable delimiters is maintained using **Domains - HIG9120** and updating Domain **CSV Delimiters**. The standard Delimiters shipped as metadata are Comma, Bar, Tab and Bang (!).

Date Mask (Optional) List

A default Date mask may be applied to the Load File definition. If applied this will be the format the loader expects for all Loaded dates. This default Date Mask may be overridden for individual Columns by specifying a different Date Mask for the individual file column.

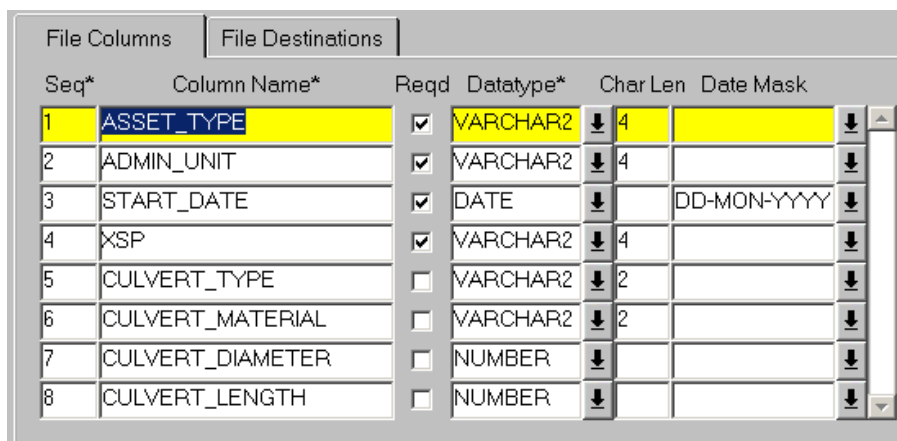
Holding Table (Required) Radio Button

If the holding table already exists on the database the select the radio button 'User Named'. You will then be required to enter the table name as shown below.



If you select User Named the 'Create Holding Table' button will be disabled. If the table doesn't exist then select 'Default' and a new table will be created for you, named NM_LD_unique_ref_TMP where unique ref is the unique ref of this table definition.

Figure 11



Seq*	Column Name*	Reqd	Datatype*	Char Len	Date Mask
1	ASSET_TYPE	<input checked="" type="checkbox"/>	VARCHAR2	4	
2	ADMIN_UNIT	<input checked="" type="checkbox"/>	VARCHAR2	4	
3	START_DATE	<input checked="" type="checkbox"/>	DATE		DD-MON-YYYY
4	XSP	<input checked="" type="checkbox"/>	VARCHAR2	4	
5	CULVERT_TYPE	<input type="checkbox"/>	VARCHAR2	2	
6	CULVERT_MATERIAL	<input type="checkbox"/>	VARCHAR2	2	
7	CULVERT_DIAMETER	<input type="checkbox"/>	NUMBER		
8	CULVERT_LENGTH	<input type="checkbox"/>	NUMBER		

File Columns Panel

This panel is used to define the Columns which will be used when creating the Holding Table for the Data Type to be loaded and should map to the data within the CSVdata file to be loaded.

Seq (Required)

Enter the sequence number for this Column within the Data file.

Column Name (Required)

Enter the name for the Column in the Holding table. The name should not include spaces.

Reqd (Checkbox)

Select this checkbox if a value is required for this field. If it is selected and a record has a NULL value for the field it will fail the Basic Data Type validation during the Load into the Holding Table.

Datatype (Required) List

Select the appropriate Datatype for the data being held in this Column. If a record has a value of the incorrect Datatype it will fail the Basic Data Type validation during the Load into the Holding Table.

CHAR Length (Optional)

Enter the maximum length of a character string allowed in this Column. If a record has a value which is longer than the specified length it will fail the Basic Data Type validation during the Load into the Holding Table.

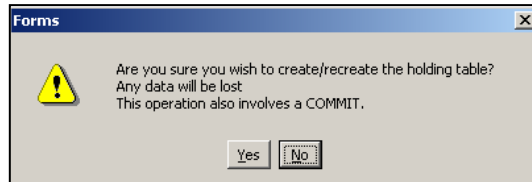
Date Mask (Optional) List

If required enter a Date Mask for the Column. If applied this will be the format the loader expects for Dates being loaded into this column. This Date Mask will take precedence over the default Date Mask if applied.

Creating a Holding Table

If you have specified a Default Holding Table then you need to create the table itself. To do this first ensure that the File Columns values have been saved (press the **[Commit]** button on the Menu Toolbar) and press the **[Create Holding Table]** button on the form. A warning message (Figure 12) will be displayed to the User warning that any data already in the Holding Table (if it has been previously created) will be lost. To continue the operation press **[Yes]** or **[No]** to cancel.

Figure 12



A message will be displayed when the table has been created. The naming convention for the Holding Table is as follows:

NM_LD_<<File Definition Unique>>_TMP

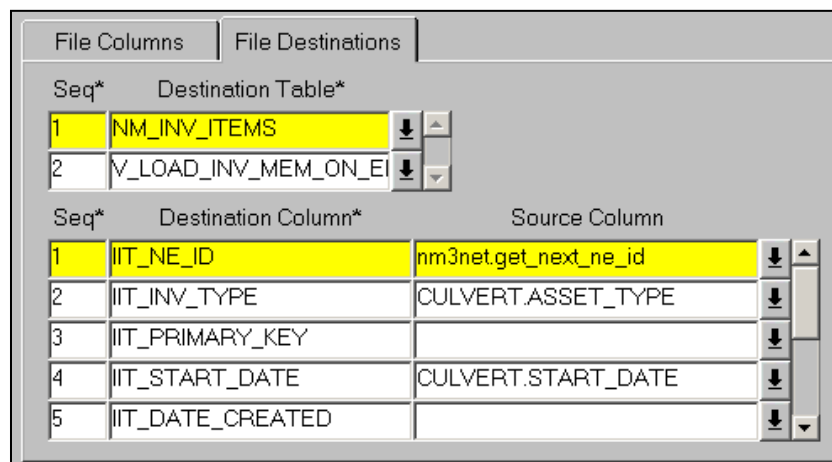
For example if the Unique of the File Definition was **LINE** the holding table would be named

NM_LD_LINE_TMP

The holding table will also contain two additional columns, namely BATCH_NO and RECORD_NO.

In the case of Asset data, if the Asset metamodel is amended, for example a new Attribute is defined for an Asset Type, a new column should be added into the File definition Table used to load Items of that Asset Type to cater for the new Attribute and the Holding Table should be recreated.

Figure 13



File Destinations Panel

The File Destinations panel is used to map the Columns in the selected Destination table with the data Source. The Source of the data will generally be from within the Holding Table, but may also be a Column within another Destination Table or a call to a database Procedure.

Each File definition may have more than one associated Destination Table. For example, when loading Asset Data a Destinations table will have been defined for the Asset item details and another for the Asset Location details. If Loading Network Nodes different Destination Tables would be specified for the Nodes and the Points.

Destination Table

Seq (Required)

Enter the sequence number which determines the order in which data is loaded into the specified Destination Table. It is important to load data in the correct order when the data Source is a Column within another Destination Table.

Destination Table (Required) List

Enter the name of the Destination Table to which the mapping relates. Destination Tables will have already been defined using the **CSV Loader Destination Table - HIG2010** module.

When the required Destination Table has been selected press the **[Save]** button on the menu toolbar. To retrieve the Columns within the selected Destination Table press the **[Execute Query]** button on the menu toolbar. The Table Columns will be displayed in the Destination Column fields along with a sequence Number.

Each of the required Destination Columns must be mapped to their data Source using the 'Source Column' field.

Note that it is not always necessary for all Columns within a Destination Table to be mapped to a data Source.

Source Column List

Enter the data Source that maps to the Destination Column. The List of Values will contain the Names of the File Columns previously specified. These will be appended with the Unique Reference of the File Definition Table. If a Default value for a Table Column has been defined in the **CSV Loader Destination Table - HIG2010** module it will be automatically populated in the appropriate 'Source Column' field. As well as the File Columns specified for the File Definition Table the data Source may be a Column from within another Destination Table. To reference a Column from within another Destination Table the required Column should be appended with the alias defined for the Destination Table in the 'Abbrev' field in the **CSV Loader Destination Table - HIG2010** module. Figure 14 shows an example where a Source Column references a Column named IIT_NE_ID which is in a Destination Table (NM_INV_ITEMS) with an Abbreviation of IIT.

Figure 14

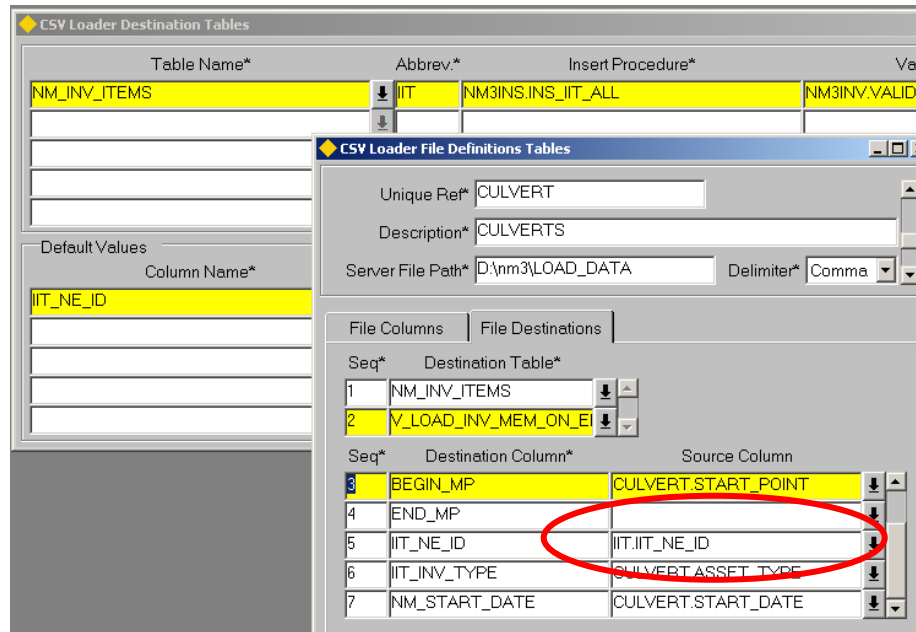


Table Name*	Abbrev.*	Insert Procedure*	Value*
NM_INV_ITEMS	IIT	NM3INS.INS_IIT_ALL	NM3INV.VALID

Seq*	Destination Table*
1	NM_INV_ITEMS
2	V_LOAD_INV_MEM_ON_E

Seq*	Destination Column*	Source Column
3	BEGIN_MP	CULVERT.START_POINT
4	END_MP	
5	IIT_NE_ID	IIT.IIT_NE_ID
6	IIT_INV_TYPE	CULVERT.ASSET_TYPE
7	NM_START_DATE	CULVERT.START_DATE

The value entered for a Source Column may also be a call to a Database procedure such as **NM3USER.GET_EFFECTIVE_DATE** which returns the effective date or **NM3NET.GET_NEXT_NE_ID** which returns the next internal Unique from a sequence when Defining the IIT_NE_ID or NE_ID when loading Asset or Network Data respectively.

A 'Constant' Value may also be added in the Source Column, For example, if loading Asset Data and the Asset Type Code was not contained within the load file, the Asset Type could be entered directly into the 'Source' column, e.g. 'SIGN', of the IIT_INV_TYPE Destination Column when using the NM_INV_ITEMS Destination Table.

Note that the Asset Type should be enclosed in single quotes as the value is of a VARCHAR2 data type

Loading Asset

The loading of Asset Data using the CSV Data Loaders will generally involve the definition of the 'Holding' or temporary Table used in the first stage of the loading process and 2 Destination Tables for the second stage Transfer and Validation of data into the appropriate tables within **Exor**. The 2 Destination Tables will cater for the Asset Items and Asset Item locations respectively. If Asset Locations are not being loaded the 'locations' Destination Table need not be defined.

This example details the File Column values needed for the Holding or temporary table and the required Destination Tables and 'column mappings' needed to load Asset and Asset Location Data for an Asset Type of '**CULV**' - Culvert.

The Culvert Asset Type is a Continuous Type and has allowable XSP values. Table 16 shows the flexible Attributes have been defined for Culvert Asset Type using the **Asset Metamodel - NM0410** module.

Table 16

Column Name	Screen Text	Data Type	Length	Dec. Places
IIT_CHR_ATTRIB26	Culvert Type	VARCHAR2	2	
IIT_CHR_ATTRIB27	Culvert Material	VARCHAR2	2	
IIT_NUM_ATTRIB16	Culvert Diameter	NUMBER	4	
IIT_NUM_ATTRIB17	Culvert Length	NUMBER	3	1

In addition to the flexible Attributes displayed in Table 16 the Data File to be loaded includes the following information for each record.

- Asset Type
- Admin Unit Name
- Start Date of Item
- XSP
- Route on which Item is located
- Start Offset
- End Offset

Figure 15 shows an extract from the Culvert load file, which is a CSV file. The data is shown in tabular format with headers added for clarity in Table 17.

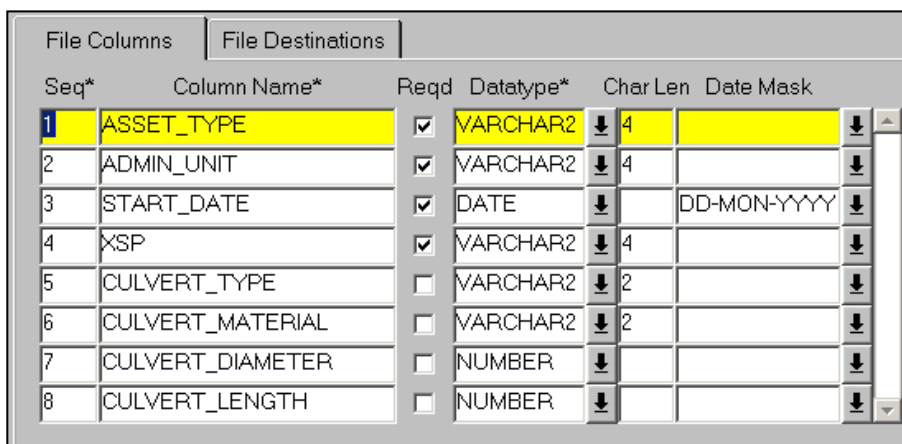
Figure 15

```
CULV,REG1,01-JAN-2000,LD,S,C,15,12.5,H26,0,5.78
CULV,REG1,01-JAN-2000,RD,S,C,10,10,H26,0,10
CULV,REG1,01-JAN-2000,LD,G,S,12,15,H26,9,15
```

Table 17

Asset Type	Admin Unit	Start Date	XSP	Culvert Type	Culvert Material	Culvert Diam.	Culvert Length	Route	Start Offset	End Offset
CULV	REG1	01-JAN-2000	LD	S	C	15	12.5	H26	0	5.78
CULV	REG1	01-JAN-2000	RD	S	C	10	10	H26	0	10
CULV	REG1	01-JAN-2000	LD	G	S	12	15	H26	9	15

Figure 16



Seq*	Column Name*	Req'd	Datatype*	Char Len	Date Mask
1	ASSET_TYPE	<input checked="" type="checkbox"/>	VARCHA2	4	
2	ADMIN_UNIT	<input checked="" type="checkbox"/>	VARCHA2	4	
3	START_DATE	<input checked="" type="checkbox"/>	DATE		DD-MON-YYYY
4	XSP	<input checked="" type="checkbox"/>	VARCHA2	4	
5	CULVERT_TYPE	<input type="checkbox"/>	VARCHA2	2	
6	CULVERT_MATERIAL	<input type="checkbox"/>	VARCHA2	2	
7	CULVERT_DIAMETER	<input type="checkbox"/>	NUMBER		
8	CULVERT_LENGTH	<input type="checkbox"/>	NUMBER		

File Column Settings

The File Columns settings for the Culvert load file are displayed in Table 18.

Table 18

Seq	Column Name	Datatype	Char Len
1	ASSET_TYPE	VARCHA2	4
2	ADMIN_UNIT	VARCHA2	4
3	START_DATE	DATE	
4	XSP	VARCHA2	4
5	CULVERT_TYPE	VARCHA2	2
6	CULVERT_MATERIAL	VARCHA2	2
7	CULVERT_DIAMETER	NUMBER	
8	CULVERT_LENGTH	NUMBER	
9	ROUTE	VARCHA2	30
10	START_OFFSET	NUMBER	
11	END_OFFSET	NUMBER	

Once the File Definition Table has been defined press the **[Save]** button on the menu toolbar followed by the **[Create Holding Table]** button on the form. This will create the Holding table for the CULVERT asset type, used in the first stage of the loading process.

File Destination Settings

As the Culvert Items are located on a Network and the location for each Item is being loaded, there are 2 Destination Tables which must be defined. The first of these is the **NM_INV_ITEMS** table which is used to store the Asset Item Details and the second table is used to store the Asset Item location. The Asset Item location Destination should be selected from one of the Destination Tables as detailed on page 8. This example uses the **V_LOAD_INV_MEM_ON_ELEMENT** Destination Table and Insert/Validation procedures supplied as standard metadata.

NM_INV_ITEMS (Asset Item Details)

Each Asset Item must have values populated for the following Columns as well as any of flexible Attributes flagged as 'Required' in the **Asset Metamodel - NM0410** module (refer to the Asset Manager System Admin Guide).

Note that if the 'XSP Allowed' flag is selected the Column within the Holding Table should also be 'mapped' to the IIT_X_SECT Column in the NM_INV_ITEMS Destination Table.

- IIT_NE_ID
- IIT_INV_TYPE
- IIT_START_DATE
- IIT_ADMIN_UNIT

IIT_NE_ID

This is the internal unique id of the Asset Item. A default value of **nm3net.get_next_ne_id** is supplied as standard metadata for the Column (page 7). This is a function call to an internal procedure which will get the next available value from the appropriate sequence.

IIT_INV_TYPE

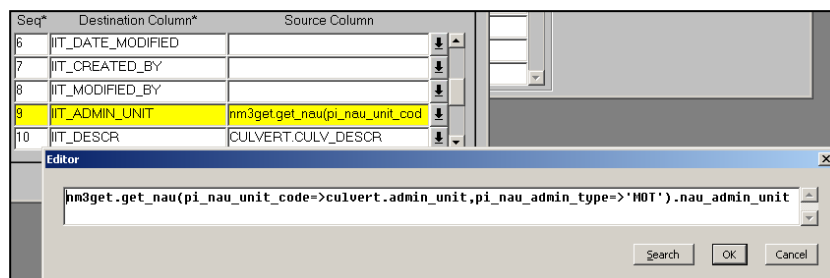
This is the Asset Type code, e.g. CULV, and would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

If the Asset Type code is not included in the load file it may be 'hard coded' into the Source Column for IIT_INV_TYPE. To do this enter the required Asset Type enclosed in single quotes, e.g. 'CULV'. Figure 17 shows an example where an Asset Type of BORD has been defined as the Asset Type.

Figure 17

Seq*	Destination Column*	Source Column	
1	IIT_NE_ID	nm3net.get_next_ne_id	↓ ↑
2	IIT_INV_TYPE	'BORD'	↓ ↑
3	IIT_PRIMARY KEY		↓ ↑

Figure 20



Seq	Destination Column	Source Column
6	IIT_DATE_MODIFIED	
7	IIT_CREATED_BY	
8	IIT_MODIFIED_BY	
9	IIT_ADMIN_UNIT	nm3get.get_nau(pi_nau_unit_cod)
10	IIT_DESCR	CULVERT.CULV_DESCR

Editor

```
nm3get.get_nau(pi_nau_unit_code=>culvert.admin_unit,pi_nau_admin_type=>'MOT').nau_admin_unit
```

Search OK Cancel

Table 19 show the complete 'mapping' of the Source Columns to the Columns in the **NM_INV_ITEMS** Destination Table (Item Details) for the CULV - Culvert Asset Type.

Table 19

Seq	Destination Column	Source Column
1	IIT_NE_ID	nm3net.get_next_ne_id
2	IIT_INV_TYPE	CULVERT.ASSET_TYPE
3	IIT_START_DATE	CULVERT.START_DATE
4	IIT_ADMIN_UNIT	nm3get.get_nau(pi_nau_unit_code=>culvert.admin_unit,pi_nau_admin_type=>'MOT').nau_admin_unit
5	IIT_NUM_ATTRIB16	CULVERT.CULVERT_DIAMETER
6	IIT_NUM_ATTRIB17	CULVERT.CULVERT_LENGTH
7	IIT_CHR_ATTRIB26	CULVERT.CULVERT_TYPE
8	IIT_CHR_ATTRIB27	CULVERT.CULVERT_MATERIAL
9	IIT_X_SECT	CULVERT.XSP

V_LOAD_INV_MEM_ON_ELEMENT (Asset Item Locations)

The following Columns within the V_LOAD_INV_MEM_ON_ELEMENT Destination Table should be 'mapped' as described below.

NE_UNIQUE

This should map to the Column within the Holding Table which stores Route or Datum Element Unique on which the Item is located.

NE_NT_TYPE

This is the Network Type of the Route or Datum Element on which the Item is located. The Source Column may be left blank for this field if the NE_UNIQUE is truly unique within the **Exor** Database. For example, if the NE_UNIQUE of a Route was H26 the NE_NT_TYPE column could remain 'unmapped' provided there was not another Route called the H26 of a different Network Type.

BEGIN_MP

This is the Start Offset of the Item and should map to the Column within the Holding Table which stores this value.

END_MP

This is the End Offset of the Item and should map to the Column within the Holding Table which stores this value. If the Item Type to which this File Definition Table relates has been 'flagged' as a Point Item Type in the **Asset Metamodel - NM0410** module (refer to the Asset Manager System Admin Guide) the END_MP Source Column may be left blank

If the Source Columns for the BEGIN_MP and END_MP Columns are left blank Continuous Asset Items loaded using this File Definition will be located over the entire specified Route.

IIT_NE_ID

This is the internal unique id of the Asset Item. A default value of **IIT.IIT_NE_ID** is supplied as standard metadata for the Destination Column (page 7). This is a 'mapping' to the **IIT_NE_ID** Column of the NM_INV_ITEMS Destination Table which has the Abbreviation of **IIT**.

IIT_INV_TYPE

This is the Asset Type code, e.g. CULV, and would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

If the Asset Type code is not included in the load file it may be 'hard coded' into the Source Column for IIT_INV_TYPE. To do this enter the required Asset Type enclosed in single quotes, e.g. 'CULV'. Figure 21 shows an example where an Asset Type of BORD has been defined as the Asset Type.

Figure 21

File Columns		File Destinations	
Seq*	Destination Table*		
1	NM_INV_ITEMS		
2	V_LOAD_INV_MEM_ON_EI		
Seq*	Destination Column*	Source Column	
3	BEGIN_MP	BORDER_CROSSING.OFFSET	
4	END_MP		
5	IIT_NE_ID	IIT.IIT_NE_ID	
6	IIT_INV_TYPE	'BORD'	
7	NM_START_DATE	BORDER_CROSSING.START_D.	

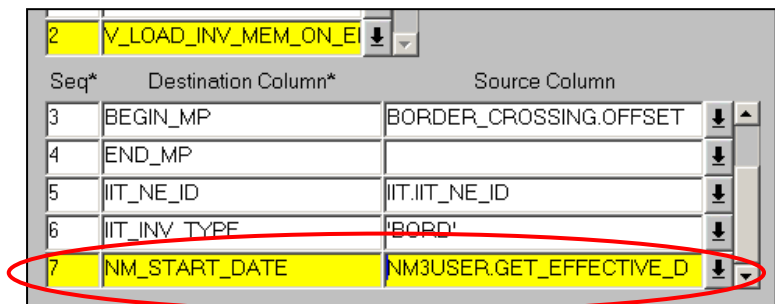
START_DATE

A Start date for both the Asset Item and the Asset Item Location (if located) is required. This value would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

Note that the same Column in the Holding Table may be 'mapped' to the Start Date in the Location Table.

If the Start Date is not included in the data file to be loaded a function call to an internal procedure may be entered which will 'derive' the start date. For example to generate the Start Date as the current effective date for each of the data Items being loaded the following function call may be used NM3USER.GET_EFFECTIVE_DATE.

Figure 22



Seq*	Destination Column*	Source Column
2	V_LOAD_INV_MEM_ON_EI	
3	BEGIN_MP	BORDER_CROSSING.OFFSET
4	END_MP	
5	IIT_NE_ID	IIT.IIT_NE_ID
6	IIT_INV_TYPE	'BORD'
7	NM_START_DATE	NM3USER.GET_EFFECTIVE_D

Loading Asset Data using Reference Post Referencing

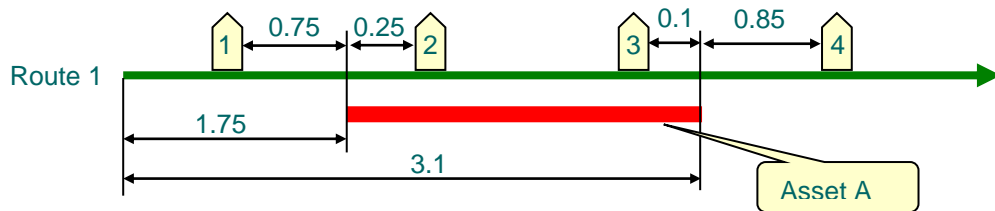
Reference Post or 'Route Mile Post' referencing, allows the location of an Asset to be expressed relative to the distance from a known 'reference' Asset, located on the selected Route. The 'Reference' Asset must be a 'Point Item' and can be either an Asset stored within the Exor database or an External Asset.

The 'Reference Post Offset' is always entered relative to the Route cardinality. Therefore a positive distance from a Reference Post means that the Asset is located further along the Route than the 'reference Post', whereas a negative distance means the Asset is located on the Route before the 'reference post'.

If no reference Item is specified, the Reference Offset is taken as the distance from the Start of the Route.

Consider the example below.

Figure 23



Asset 'A' could be located in several different ways relative to the Marker Posts (MP) shown in the diagram. Its start location could be expressed as follows.

Route	MP	Offset
1	1	0.75
Or		
Route	MP	Offset
1	2	-0.25
Or		
Route	MP	Offset
1	null	1.75

Its end location could be expressed as follows:

Route	MP	Offset
1	3	0.1
Or		
Route	MP	Offset
1	4	-0.85
Or		
Route	MP	Offset
1	null	3.1

This example details the File Column values needed for the Holding or temporary table and the required Destination Tables and 'column mappings' needed to load Asset and Asset Location Data using Reference Post Referencing for an Asset Type of '**SIGN**' – Sign Post. An Asset Type of 'MP' Mile Post' is being used as the Reference Post Type.

The Sign Asset Type is a Point Type. The table below shows the flexible Attributes have been defined for Sign Asset Type using the **Asset Metamodel - NM0410** module.

Table 20

Column Name	Screen Text	Data Type	Length	Dec. Places
IIT_CHR_ATTRIB26	Sign Type	VARCHAR2	10	

In addition to the flexible Attributes displayed in Table 20 the Data File to be loaded includes the following information for each record.

- Asset Type
- Sign Type
- Route on which Item is located
- Reference Post Type for Start Location
- Reference Post Item Label (iit_descr) for Start Location
- Offset from Reference Post for Start Location

Note that for continuous Items, the data file must also contain the following:

- Reference Post Type for End Location
- Reference Post Item Label (iit_descr) for End Location
- Offset from Reference Post for End Location

Figure 24 shows an extract from the Sign load file, which is a CSV file. The data is shown in tabular format with headers added for clarity in Table 21.

Figure 24

```

SIGN,STOP,BIA-AIA013-P,,,0.2
SIGN,STOP,BIA-AIA013-P,MP,MP - 1301929,0.2
SIGN,GIVE WAY,BIA-AIA013-P,MP,MP - 1301930,0.25
SIGN,STOP,BIA-AIA013-P,MP,MP - 1301932,0.8
SIGN,GIVE WAY,BIA-AIA013-P,MP,MP - 1301948,-0.3

```

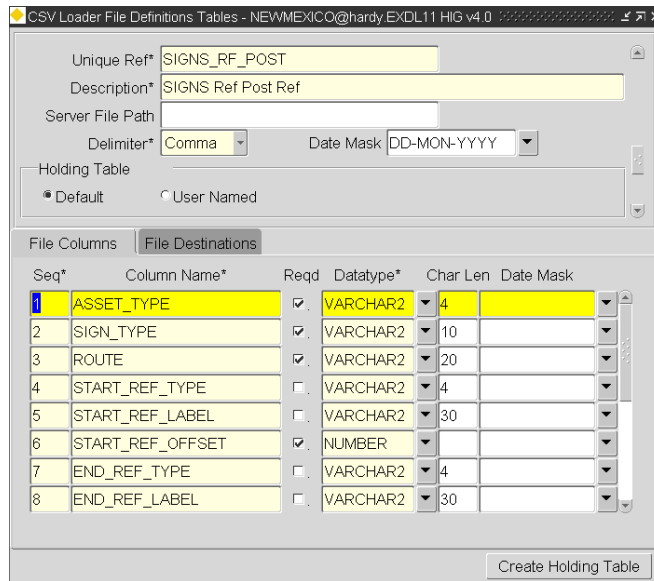
Table 21

Asset Type	Sign Type	Route	Ref Post Type	Ref Post Item Label	Ref Post Offset
SIGN	STOP	BIA-AIA013-P	NULL	NULL	0.2
SIGN	STOP	BIA-AIA013-P	MP	MP - 1301929	0.2
SIGN	GIVE WAY	BIA-AIA013-P	MP	MP - 1301930	0.25
SIGN	STOP	BIA-AIA013-P	MP	MP - 1301932	0.8
SIGN	GIVE WAY	BIA-AIA013-P	MP	MP - 1301948	-0.3

Asset will be loaded 0.2 from the start of the route

Negative Ref Post Offset

Figure 25



File Column Settings

The File Columns settings for the SIGN load file are displayed in Table 22

Table 22

Seq	Column Name	Datatype	Char Len
1	ASSET_TYPE	VARCHAR2	4
2	SIGN_TYPE	VARCHAR2	10
3	ROUTE	VARCHAR2	20
4	XSP	VARCHAR2	4
5	START_REF TYPE	VARCHAR2	4
6	START_REF LABEL	VARCHAR2	30
7	START_REF_OFFSET	NUMBER	

Once the File Definition Table has been defined press the **[Save]** button on the menu toolbar followed by the **[Create Holding Table]** button on the form. This will create the Holding table for the SIGN asset type, used in the first stage of the loading process.

File Destination Settings

As the SIGN Items are located on a Network and the location for each Item is being loaded, there are 2 Destination Tables which must be defined. The first of these is the **NM_INV_ITEMS** table which is used to store the Asset Item Details and the second table is used to store the Asset Item location. The Asset Item location Destination should be selected from one of the Destination Tables as detailed on page 8. This example uses the **V_LOAD_LOCATE_INV_BY_REF** Destination Table and Insert procedure supplied as standard metadata.

NM_INV_ITEMS (Asset Item Details)

Each Asset Item must have values populated for the following Columns as well as any of flexible Attributes flagged as 'Required' in the **Asset Metamodel - NM0410** module (refer to the Asset Manager System Admin Guide).

- IIT_NE_ID
- IIT_INV_TYPE
- IIT_START_DATE
- IIT_ADMIN_UNIT

IIT_NE_ID

This is the internal unique id of the Asset Item. A default value of **nm3net.get_next_ne_id** is supplied as standard metadata for the Column (page 7). This is a function call to an internal procedure which will get the next available value from the appropriate sequence.

IIT_INV_TYPE

This is the Asset Type code, e.g. SIGN, and would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

If the Asset Type code is not included in the load file it may be 'hard coded' into the Source Column for IIT_INV_TYPE. To do this enter the required Asset Type enclosed in single quotes, e.g. 'SIGN'. Figure 26 shows an example where an Asset Type of BORD has been defined as the Asset Type.

Figure 26

Seq*	Destination Column*	Source Column	
1	IIT_NE_ID	nm3net.get_next_ne_id	
2	IIT_INV_TYPE	'BORD'	
3	IIT_PRIMARY_KEY		

START_DATE

A Start date for both the Asset Item and the Asset Item Location (if located) is required. This value would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

Note that the same Column in the Holding Table may be 'mapped' to the Start Date in the Location Table.

If the Start Date is not included in the data file to be loaded a function call to an internal procedure may be entered which will 'derive' the start date. For example to generate the System Date as the current effective date for each of the data Items being loaded the following function call may be used `NM3USER.GET_EFFECTIVE_DATE`. Figure 27 shows an example of this in use.

Figure 27

Seq*	Destination Column*	Source Column
1	IIT_NE_ID	NM3NET.GET_NEXT_NE_ID
2	IIT_INV_TYPE	SIGNS_RF_POST.ASSET_TYPE
3	IIT_PRIMARY_KEY	NM3NET.GET_NEXT_NE_ID
4	IIT_START_DATE	NM3USER.GET_EFFECTIVE_DATE
5	IIT_DATE_CREATED	

IIT_ADMIN_UNIT

This is the Admin Unit of the Asset Item. The Insert procedures used when transferring the data from the Holding Table to the Destination Table expects the internal unique of the Admin Unit to be supplied. Since this value is unlikely to be known it is more common for the Admin Unit name to be supplied for each record within the data file to be loaded. This Admin Unit name may be 'decoded' to the internal Admin Unit id using a procedure which should be entered in the Source Column for the IIT_ADMIN_UNIT Destination Column. An example is shown below.

```
nm3get.get_nau(pi_nau_unit_code=>SIGN_RF_POST.admin_unit,pi_nau_admin_type=>'MOT').nau_admin_unit
```

where **SIGN_RF_POST** is the Unique Ref (Figure 25) of the current File Definition Table and **MOT** is the Admin Type Code of the Admin Unit

Note that this will be the Admin Type associated with Asset Type in the Asset Metamodel - NM0410 module.

Table 23 show the complete 'mapping' of the Source Columns to the Columns in the **NM_INV_ITEMS** Destination Table (Item Details) for the SIGN - SIGN Asset Type.

Table 23

Seq	Destination Column	Source Column
1	IIT_NE_ID	nm3net.get_next_ne_id
2	IIT_INV_TYPE	SIGNS_RF_POST.ASSET_TYPE
3	IIT_START_DATE	NM3USER.GET_EFFECTIVE_DATE
4	IIT_ADMIN_UNIT	7 (this has been hardcoded or this example)
5	IIT_NUM_ATTRIB16	CULVERT.CULVERT_DIAMETER
7	IIT_CHR_ATTRIB26	CULVERT.CULVERT_TYPE

V_LOAD_LOCATE_INV_BY_REF (Locate Assets using Reference Post Referencing)

The following Columns within the V_LOAD_LOCATE_INV_BY_REF Destination Table should be 'mapped' as described below.

IIT_NE_ID

This is the internal unique id of the Asset Item. A default value of **IIT.IIT_NE_ID** is supplied as standard metadata for the Destination Column (page 7). This is a 'mapping' to the **IIT_NE_ID** Column of the NM_INV_ITEMS Destination Table which has the Abbreviation of **IIT**.

EFFECTIVE_DATE

A Start date for both the Asset Item and the Asset Item Location (if located) is required. This value would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

Note that the same Column in the Holding Table may be 'mapped' to the Start Date in the Location Table.

If the Start Date is not included in the data file to be loaded a function call to an internal procedure may be entered which will 'derive' the start date. For example to generate the Start Date as the current effective date for each of the data Items being loaded the following function call may be used **NM3USER.GET_EFFECTIVE_DATE**.

ROUTE_ID

This is the internal NE_ID of the Route on which the Asset is located. The Insert procedure used when transferring the data from the Holding Table to the Destination Table expects the internal ne_id of the Route to be supplied. Since this value is unlikely to be known it is more common for the Unique (ne_unique) name to be supplied for each record within the data file to be loaded. This Route name may be 'decoded' to the internal ne_id using a procedure which should be entered in the Source Column for the ROUTE_ID Destination Column. An example is shown below.

```
nm3get.get_ne_all(pi_ne_unique=>signs_rf_post.ROUTE,pi_ne_nt_type=>'ROUT').ne_id
```

where **SIGN_RF_POST** is the Unique Ref (Figure 25) of the current File Definition Table, **ROUTE** is the File Column which stores the Route Unique and **ROUT** is the Network Type of the Route.

START_REF_TYPE

This the Asset Type of the Reference Post Asset Type used for the Start Offset. This may be hardcoded with the appropriate Asset Type, e.g. 'MP', or mapped to Column within the Holding Table which stores this value.

Note that if a null value is passed in for the START_REF_TYPE and START_REF_LABEL , the START_REF_OFFSET value will be relative to the start of the Route

START_REF_LABEL

This is the Reference Post item against which the Asset is being located. The IIT_DESCR of the reference item must be used.

Note that if a null value is passed in for the START_REF_TYPE AND START_RF_LABEL , the START_REF_OFFSET value will be relative to the start of the Route

START_REF_OFFSET

This is the distance from the selected Reference Post for the Start of the Assets. A positive values means that the Asset is located further along the Route than the Reference Post. A negative value means it is located on the Route before the 'reference post'.

END_REF_TYPE

This the Asset Type of the Reference Post Asset Type usEd for the End Offset. This may be hardcoded with the appropriate Asse Type, e.g. 'MP', or mapped to Column within the Holding Table which stores this value.

This column should be left blank when loading Point Asset Types.

Note that if a null value is passed in for the END_REF_TYPE and END_REF_LABEL , the END_REF_OFFSET value will be relative to the start of the Route

END_REF_LABEL

This is the Reference Post item against which the Asset is being located. The IIT_DESCR of the reference item must be used.

Note that if a null value is passed in for the END_REF_TYPE and END_REF_LABEL , the END_REF_OFFSET value will be relative to the start of the Route

This column should be left blank when loading Point Asset Types.

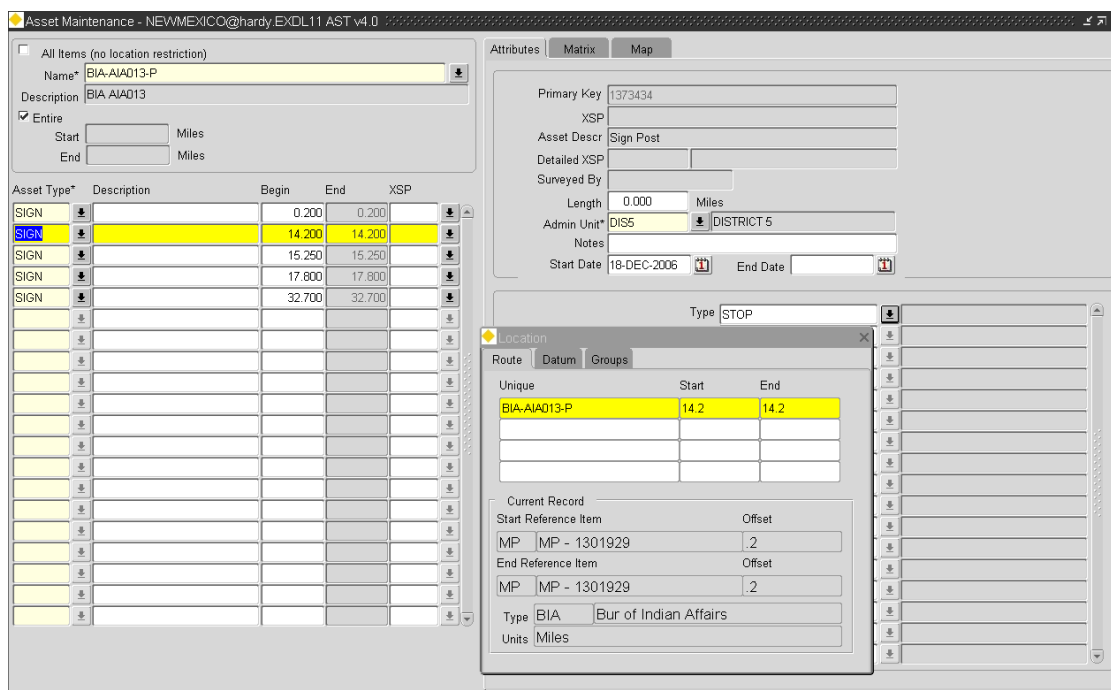
END_REF_OFFSET

This is the distance from the selected Reference Post for the End of the Asset. A positive values means that the Asset is located further along the Route than the Reference Post. A negative value means it is located on the Route before the 'reference post'.

This column should be left blank when loading Point Asset Types.

Figure 28 shows the load Signs within the **Asset Maintenance – NM0590** module.

Figure 28



The screenshot shows the 'Asset Maintenance' window for 'NEWMEXICO@hardy EXDL11 AST v4.0'. It features a table of assets and a detailed view of a selected asset.

Asset Type*	Description	Begin	End	XSP
SIGN		0.200	0.200	
SIGN		14.200	14.200	
SIGN		15.250	15.250	
SIGN		17.800	17.800	
SIGN		32.700	32.700	

The detailed view for asset 'BIA-AIA013-P' shows the following information:

- Primary Key: 1373434
- XSP: Sign Post
- Detailed XSP:
- Surveyed By:
- Length: 0.000 Miles
- Admin Unit: DISTRICT 5
- Notes:
- Start Date: 18-DEC-2006
- End Date:

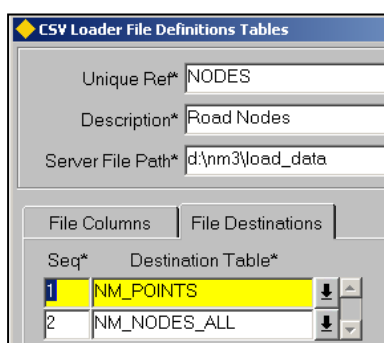
A 'Location' dialog box is also open, showing a table with columns: Unique, Start, End. The table contains one row: BIA-AIA013-P, 14.2, 14.2.

Loading Network Nodes

The loading of Network Nodes using the CSV Data Loaders will involve the definition of the 'Holding' or temporary Table used in the first stage of the loading process and 2 Destination Tables for the second stage Transfer and Validation of data into the appropriate tables within **Exor**. The 2 Destination Tables will cater for **NM_POINTS** table (in which the Grid Coordinates are held) and the **NM_NODES_ALL** table respectively.

When defining the Column Mappings in the File Destination Tab, the **NM_POINTS** table **MUST** be sequenced before the **NM_NODES_ALL** table as shown in Figure 29.

Figure 29



The screenshot shows the 'CSV Loader File Definitions Tables' window. It contains the following fields:

- Unique Ref*: NODES
- Description*: Road Nodes
- Server File Path*: d:\nm3\load_data

The 'File Columns' tab is selected, showing a table with columns: Seq*, Destination Table*. The table contains two rows:

Seq*	Destination Table*
1	NM_POINTS
2	NM_NODES_ALL

This example details the File Column values needed for the Holding or temporary table and the required Destination Tables and 'column mappings' needed to load Network Nodes.

Figure 30 shows an extract from the Nodes load file, which is a CSV file. The data is shown in tabular format with headers added for clarity below.

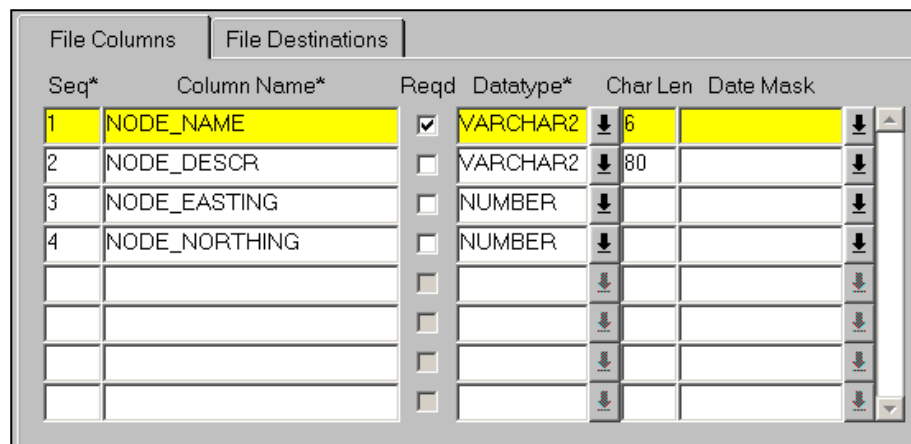
Figure 30

```
000001,Int Church Avenue and Main Street,577436,1684831
000002,Rte H26 and I52 East,580827,1682981
000003,Int Church Avenue and Glebe Road,580981,1680516
000004,Int Hillsborough Road and Glebe Road,583447,1680824
000005,Glebe Road Int with Back Lane,583909,1679899
000006,Int Harrys Road and A1,581597,1681594
```

Table 24

Node Name	Description	Easting	Northing
000001	Int Church Avenue and Main Street	577436	1684831
000002	Rte H26 and I52 East	580827	1682981
000003	Int Church Avenue and Glebe Road	580981	1680516
000004	Int Hillsborough Road and Glebe Road	583447	1680824
000005	Glebe Road Int with Back Lane	583909	1679899
000006	Int Harrys Road and A1	581597	1681594

Figure 31



Seq*	Column Name*	Req'd	Datatype*	Char Len	Date Mask
1	NODE_NAME	<input checked="" type="checkbox"/>	VARCHA2	6	
2	NODE_DESCR	<input type="checkbox"/>	VARCHA2	80	
3	NODE_EASTING	<input type="checkbox"/>	NUMBER		
4	NODE_NORTHING	<input type="checkbox"/>	NUMBER		
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			
		<input type="checkbox"/>			

File Column Settings

The File Columns settings for the Nodes load file are displayed in Table 25.

Table 25

Seq	Column Name	Datatype	Char Len
1	NODE_NAME	VARCHA2	6
2	NODE_DESCR	VARCHA2	80
3	NODE_EASTING	NUMBER	
4	NODE_NORTHING	NUMBER	

Once the File Definition Table has been defined press the **[Save]** button on the menu toolbar followed by the **[Create Holding Table]** button on the form. This will create the Holding table for the Node load file, used in the first stage of the loading process.

File Destination Settings

When loading Network Nodes there are 2 File Destination Tables to be 'mapped' as explained on page (36). The **NM_POINTS** Destination Table **MUST** be sequenced before the **NM_NODES_ALL** table.

NM_POINTS

The following Columns within the NM_POINTS Destination Table should be 'mapped' as described below. Columns not included in this description, e.g. NP_DATE_CREATED are automatically populated on insertion of a record to the table.

NP_ID (Required)

This is the internal unique id of the Point. A default value of **nm3seq.next_np_id_seq** is supplied as standard metadata for the Column (page 36). This is a function call to an internal procedure which will get the next available value from the appropriate sequence.

NP_GRID_EAST

This is the Easting Coordinate for the Point. This Column should be mapped to the appropriate Column in the Holding Table if Grid Coordinates are being loaded.

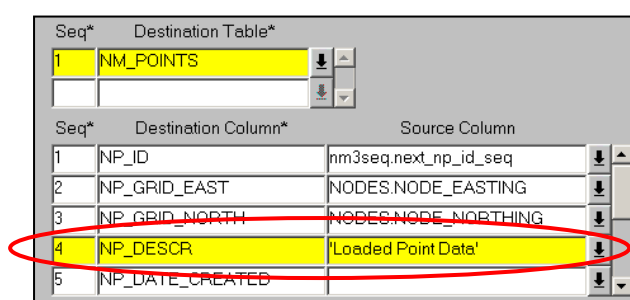
NP_GRID_NORTH

This is the Northing Coordinate for the Point. This Column should be mapped to the appropriate Column in the Holding Table if Grid Coordinates are being loaded.

NP_DESCR (Required)

This is the description of the Point. This description may be mapped to the Node Description or Node Name column within the Nodes Holding table or a constant value may be added for all Node Point descriptions by entering a text string enclosed in single quotes, e.g. 'Loaded Point Data'. Figure 32 shows an example of this in use.

Figure 32



Seq*	Destination Table*	Seq*	Destination Column*	Source Column
1	NM_POINTS	1	NP_ID	nm3seq.next_np_id_seq
		2	NP_GRID_EAST	NODES.NODE_EASTING
		3	NP_GRID_NORTH	NODES.NODE_NORTHING
		4	NP_DESCR	'Loaded Point Data'
		5	NP_DATE_CREATED	

Table 26 shows the complete 'mapping' of the Source Columns to the Columns in the **NM_POINTS** Destination Table used in this example.

Table 26

Destination Column	Source Column
NP_ID	nm3seq.next_np_id_seq
NP_GRID_EAST	NODES.NODE_EASTING
NP_GRID_NORTH	NODES.NODE_NORTHING
NP_DESCR	NODES.NODE_DESCR

NM_NODES_ALL

The following Columns within the **NM_NODES_ALL** Destination Table should be 'mapped' as described below. Columns not included in this description, e.g. NO_DATE_CREATED are automatically populated on insertion of a record to the table.

NO_NODE_ID (Required)

This is the internal unique id of the Node. It is normally set equal to the NP_ID value of the associated POINT. A default value for the Source Column of **np.np_id** is supplied as standard metadata. This is a 'mapping' to the **np_id** Column of the NM_POINTS Destination Table which has the Abbreviation of **NP**.

NO_NODE_NAME (Required)

This is Node Name visible to the User and should be mapped to the appropriate Column in the Holding Table.

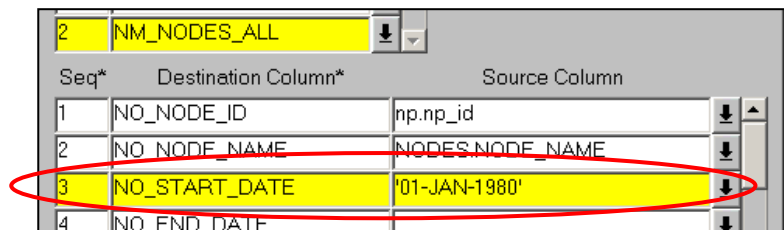
NO_START_DATE (Required)

This is the Start date of the Node. If this is included in the Node file to be loaded it should be mapped to the appropriate Column in the Holding Table.

If the Start Date is not included in the data file to be loaded a function call to an internal procedure may be entered which will 'derive' the start date. For example to generate the Start Date as the current effective date for each of the Nodes being loaded the following function call may be used **NM3USER.GET_EFFECTIVE_DATE**.

Alternatively a constant value may be added for all Node Point Start Dates by entering the required Date enclosed in single quotes, e.g. '01-JAN-1980'. Figure 33 shows an example of this in use.

Figure 33



Seq*	Destination Column*	Source Column
1	NO_NODE_ID	np.np_id
2	NO_NODE_NAME	NODES.NODE_NAME
3	NO_START_DATE	'01-JAN-1980'
4	NO_END_DATE	

NO_NP_ID (Required)

This is the internal unique of the associated POINT. A default value for the Source Column of **np.np_id** is supplied as standard metadata. This is a 'mapping' to the **np_id** Column of the NM_POINTS Destination Table which has the Abbreviation of **NP**.

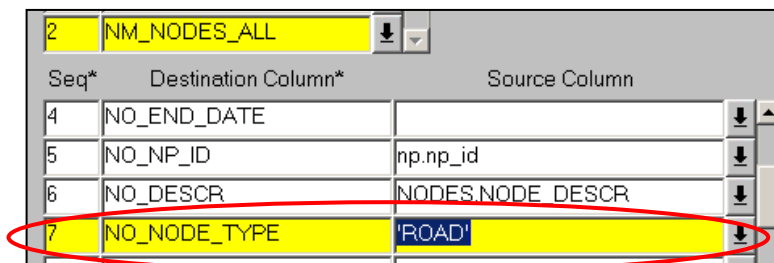
NO_DESCR (Required)

This is the description of the Node. This should be mapped to the appropriate Column in the Holding Table. If the Node Descriptions are not held within the Node load file or a constant value may be added for all Node descriptions by entering a text string enclosed in single quotes, e.g. 'Loaded Nodes'

NO_NODE_TYPE (Required)

This is the Node Type of the Nodes being loaded as defined in **Node Types - NM0001** (refer to Network Manager System Admin Guide). If the Node Type is not included in the Nodes load file it may be added as a constant value in the Source Column. The Node Type must be enclosed in single quotes. Figure 34 show an example of this in use.

Figure 34



Seq*	Destination Column*	Source Column
4	NO_END_DATE	
5	NO_NP_ID	np.np_id
6	NO_DESCR	NODES.NODE_DESCR
7	NO_NODE_TYPE	'ROAD'
8	NO_DATE_CREATED	

Table 27 shows the complete 'mapping' of the Source Columns to the Columns in the **NM_NODES_ALL** Destination Table used in this example

Table 27

Destination Column	Source Column
NO_NODE_ID	np.np_id
NO_NODE_NAME	NODES.NODE_NAME
NO_START_DATE	'01-JAN-1980'
NO_NP_ID	np.np_id
NO_DESCR	NODES.NODE_DESCR
NO_NODE_TYPE	'ROAD'

Loading Network Elements (Datum or Groups)

Datum Network Elements and Network Groups may both be loaded in a similar manner using the **NM_ELEMENTS_ALL** Destination table and the **NM3NET.INSERT_ANY_ELEMENT** insert procedure. These are supplied as standard metadata. Group members may be loaded using the **NM_MEMBERS_ALL** Destination Table as detailed on page 47.

Elements which have had Auto-inclusion records defined will be included as members of their Parent Group(s) during the loading of the Element Item. If the Auto-inclusion has also been flagged as 'Auto-Create' the loading process will also create the 'parent' Group if it does not already exist.

For more information on Auto-inclusion and Auto-creation refer to the Network Manager System Admin Guide.

During the loading of Network Elements (Datum or Groups) data is validated against the Network and Group Type Metamodel defined using the **Network Types - NM0002** and **Group Types - NM0004** modules respectively. This includes validation on the provision or otherwise of the NE_UNIQUE value when the 'Pop Unique' flag is selected for a Network Type, the Admin Unit is of the appropriate Admin Type and the values loaded for any Network Attributes are valid.

File Columns

The File Column values and the subsequent Holding Table created for loading Network Elements is determined by the Network Model defined within the **Network Types - NM0002** module. In general a Column should be defined for each of the Flexible Network Attributes defined for a Network Type plus the following 'fixed' attributes.

- NE_UNIQUE (only needed if 'Pop-Unique' flag is not selected)
- NE_DESCR
- NE_START_DATE
- NE_LENGTH (if loading Datum Elements)
- NE_ADMIN_UNIT
- NE_NO_START (Start Node, if required)
- NE_NO_END (End Node, if required)

Note that if the value for a flexible Attribute is being generated from a Database sequence (defined in the Seq Name column of the Type Columns panel of Network Types - NM0002) or derived as a result of the Attribute being used in the Control Column of an Autoinclusion record, the Attribute must not be defined within the Holding Table and a value for the Attribute should not be included within the data file.

Note on NE_SUB_CLASS Attribute

If a Datum Network Type is not using the NE_SUB_CLASS attribute as a Carriageway Identifier and the Attribute has been added purely as a System requirement, it should not be defined within the Holding Table. A default Value will have been defined for the Attribute on the *Network Type - NM0002* module. This default value will be loaded into the Destination table automatically during the load process.

File Destinations

The following Columns within the **NM_ELEMENTS_ALL** Destination Table should be 'mapped' as described below. Columns not included in this description, e.g. NE_DATE_CREATED are automatically populated on insertion of a record to the table.

NE_ID (Required)

This is the internal unique id of the Element. A default value of ***nm3seq.next_NE_ID_SEQ*** is supplied as standard metadata for the Column (page 15). This is a function call to an internal procedure which will get the next available value from the appropriate sequence.

NE_UNIQUE

This is the NE_UNIQUE of the Element. This value does not need to be loaded if the Network Type has been flagged as 'Pop Unique' but may be if so desired. If the NE_UNIQUE is included within the load file this Column should be mapped to the appropriate Column in the Holding Table.

NE_TYPE (Required)

This is the internal Network Type indicator of the Network Type and would not generally be included in the load file. A constant value may be entered into the Source Column for this field. The value should be enclosed in single quotes. The valid NE_TYPE values are :

- S** Datum Elements
- G** Groups of Sections
- P** Groups of Groups

Figure 35 shows an example where Groups of Sections are being loaded. The NE_TYPE Destination Column has been set to a constant value of 'G'.

Figure 35

File Columns		File Destinations	
Seq*	Destination Table*		
1	NM_ELEMENTS_ALL		
Seq*	Destination Column*	Source Column	
1	NE_ID	nm3seq.next_NE_ID_SEQ	
2	NE_UNIQUE		
3	NE_TYPE	'G'	
4	NE_NT_TYPE	'RFI'	

NE_NT_TYPE (Required)

This is the Network Type of the Datum Element or the Network Type associated with the Group Type being loaded and would not generally be included in the load file. If the NE_NT_TYPE is included within the load file this Column should be mapped to the appropriate Column in the Holding Table. If the Network Type is not included within the load file it may be added as a constant value into the Source. The value should be enclosed in single quotes, e.g. 'RFI'.

Figure 36 shows an example where Groups of Network Type RFI are being loaded.

Figure 36

Seq*		Destination Table*	
1		NM_ELEMENTS_ALL	
Seq*	Destination Column*	Source Column	
1	NE_ID	nm3seq.next_NE_ID_SEQ	
2	NE_UNIQUE		
3	NE_TYPE	'G'	
4	NE_NT_TYPE	'RFI'	

NE_DESCR (Required)

This is the Description of the Network Element. This should be mapped to the appropriate Column in the Holding Table.

NE_LENGTH (Datum Elements only)

This is the length of the Datum Element. This should be mapped to the appropriate Column in the Holding Table.

NE_ADMIN_UNIT

This is the Admin Unit of the Network Element. The Insert procedures used when transferring the data from the Holding Table to the Destination Table expects the internal unique of the Admin Unit to be supplied. Since this value is unlikely to be known it is more common for the Admin Unit name to be supplied for each record within the data file to be loaded. This Admin Unit name may be 'decoded' to the internal Admin Unit id using a procedure which should be entered in the Source Column for the IIT_ADMIN_UNIT Destination Column. An example is shown below.

```
nm3get.get_nau(pi_nau_unit_code=>rfl_groups.admin_unit,pi_nau_admin_type=>'MOT').nau_admin_unit
```

where **rfl_groups** is the Unique Ref (Figure 37) of the current File Definition Table and **MOT** is the Admin Type Code of the Admin Unit

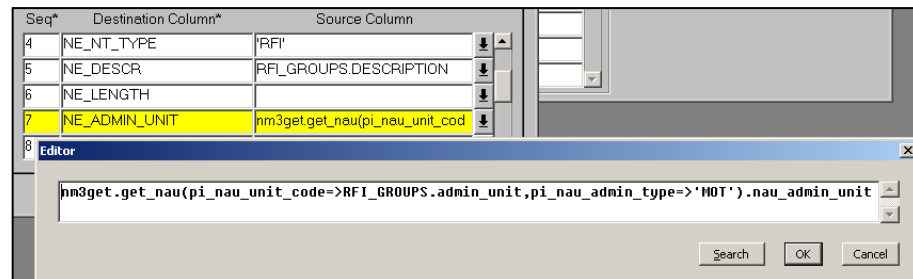
Note that this will be the Admin Type associated with Network Type in the Network Types - NM0002 module.

Figure 38 shows an example of this in use with the complete syntax displayed in the Editor.

Figure 37



Figure 38



NE_START_DATE

This is the Start date of the Network Elements. If this is included in the load file it should be mapped to the appropriate Column in the Holding Table.

If the Start Date is not included in the data file to be loaded a function call to an internal procedure may be entered which will 'derive' the start date. For example to generate the Start Date as the current effective date for each of the Nodes being loaded the following function call may be used
NM3USER.GET_EFFECTIVE_DATE.

Alternatively a constant value may be added for all Element Start Dates by entering the required Date enclosed in single quotes, e.g. '01-JAN-1980'.

Figure 39 shows an example of this in use.

Figure 39

Seq*	Destination Table*	
1	NM_ELEMENTS_ALL	
Seq*	Destination Column*	Source Column
9	NE_DATE_MODIFIED	
10	NE_MODIFIED_BY	
11	NE_CREATED_BY	
12	NE_START_DATE	'01-JAN-1980'
13	NE_END_DATE	

NE_GTY_GROUP_TYPE (Groups only)

This is the Group Type of the Groups being loaded and would not generally be included in the load file. If the Group Type is included within the load file this Column should be mapped to the appropriate Column in the Holding Table. If it is not included within the load file it may be added as a constant value into the Source. The value should be enclosed in single quotes, e.g. 'GRFI'. Figure 40 shows an example where Groups of Group Type GRFI are being loaded.

Figure 40

Seq*	Destination Table*	
1	NM_ELEMENTS_ALL	
Seq*	Destination Column*	Source Column
13	NE_END_DATE	
14	NE_GTY_GROUP_TYPE	'GRFI'
15	NE_OWNER	DEL GROUPS DIRECTION

NE_NO_START and NE_NO_END

This is the internal Node id of the Start Node of the Element. Nodes are generally associated with Datum Elements but may also be associated with Groups if modelled as so in the **Network Types - NM0002** module (refer to the Network Manager System Admin Guide).

The Insert procedures used when transferring the data from the Holding Table to the Destination Table expects the internal id of the Node Point to be supplied. Since this value is unlikely to be known it is more common for the Node Name to be supplied for each record within the data file to be loaded. This Node Name may be 'decoded' to the internal Node id using a procedure which should be entered in the Source Column for the NE_NO_START and NE_NO_END Destination Column. An example is shown below.

```
nm3get.get_no(pi_no_node_name=>RFID_DATUM.start_node,
pi_no_node_type=>'ROAD').no_node_id
```

where **RFID_DATUM** is the Unique Ref (Figure 41) of the current File Definition Table, **ROAD** is the Node Type (**note that this will be the Node Type associated with Network Type in the Network Types -**

NM0002 module) and **start_node** is the Column within the Holding Table which is used to hold the Start Node of the Element.

Figure 42 shows an example of this in use for the Start Node of a Network Element with the complete syntax displayed in the Editor.

Figure 41

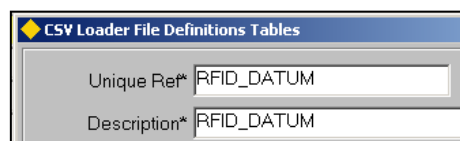
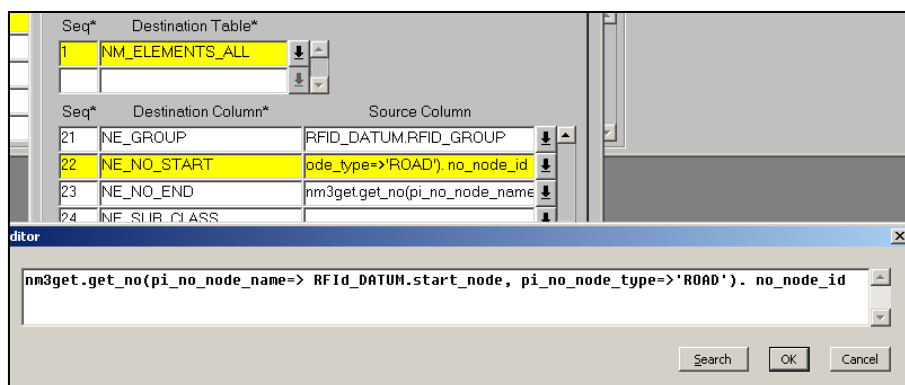


Figure 42



Seq*	Destination Table*	Source Column
1	NM_ELEMENTS_ALL	
21	NE_GROUP	RFID_DATUM.RFID_GROUP
22	NE_NO_START	nm3get.get_no(pi_no_node_name=>'ROAD'). no_node_id
23	NE_NO_END	nm3get.get_no(pi_no_node_name=> RFID_DATUM.start_node, pi_no_node_type=>'ROAD'). no_node_id
24	INF_SIR_CLASS	

Editor

```
nm3get.get_no(pi_no_node_name=> RFID_DATUM.start_node, pi_no_node_type=>'ROAD'). no_node_id
```

Flexible Attribute Columns

The Columns defined as Flexible Network Attributes should be mapped to the appropriate Column within the Holding Table.

Note that if the value for a flexible Attribute is being generated from a Database sequence (defined in the Seq Name column of the Type Columns panel of Network Types - NM0002) or derived as a result of the Attribute being used in the Control Column of an Autoinclusion record, a value for the Attribute should not be included within the data file and the Attribute must not be defined within the Holding Table.

Attribute Formatting

If a Format mask has been defined for a Network Attribute in the 'Format' field of the **Network Types - NM0002** module the loaded data will be formatted accordingly. For example, if a Format of '00000' has been defined for an Attribute and a value of '10' existed within the load file, it would be formatted and loaded as '00010'.

Default Attribute Values

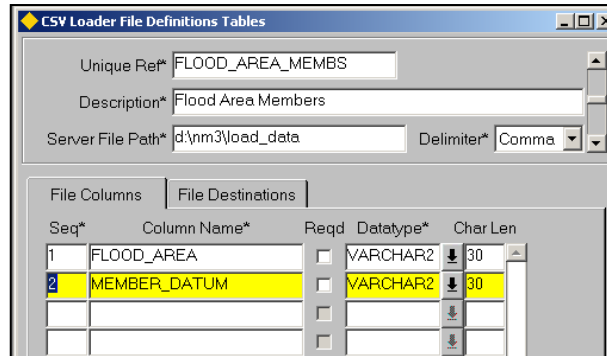
If a Default value has been added for a Network Attribute in the 'Default' field of the **Network Types - NM0002** module (and a Column has been defined for the Attribute within the Holding Table) it will be inserted into the Attribute for a record during the load process if the load file does not contain a value for the Attribute.

Loading Group Members

Members of Network Groups may be loaded using the NM_MEMBERS_ALL Destination Table and NM3INS.INS_NM_ALL insert procedure combination. This applies to loading Datum Elements or Distance Breaks into Groups of Elements or loading Groups into other Network Groups. The Elements (Datum Elements or Distance Breaks) or Groups must already exist within the **Exor Database**.

The following Figures show several possible examples.

Figure 43

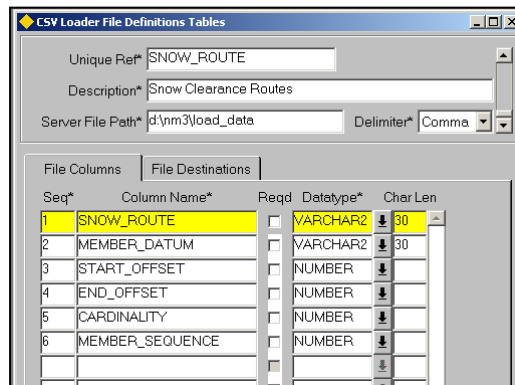


The screenshot shows the 'CSV Loader File Definitions Tables' window. The 'Unique Ref*' field is 'FLOOD_AREA_MEMBS' and the 'Description*' is 'Flood Area Members'. The 'Server File Path*' is 'd:\nm3\load_data' and the 'Delimiter*' is 'Comma'. The 'File Columns' tab is active, showing a table with the following columns:

Seq*	Column Name*	Reqd	Datatype*	Char Len
1	FLOOD_AREA	<input type="checkbox"/>	VARCHAR2	30
2	MEMBER_DATUM	<input type="checkbox"/>	VARCHAR2	30

Figure 43 shows an example where member Elements are being added for a non linear, non-partial Group. The only data being loaded into the Holding Table is the Parent Group Unique (FLOOD_AREA) and the member Element Unique (MEMBER_DATUM). Required data such as the Start Date and Start / End Offsets are either added as constant values or derived using function calls in the Source Column of the 'File Destinations' panel as described in the next section.

Figure 44



The screenshot shows the 'CSV Loader File Definitions Tables' window. The 'Unique Ref*' field is 'SNOW_ROUTE' and the 'Description*' is 'Snow Clearance Routes'. The 'Server File Path*' is 'd:\nm3\load_data' and the 'Delimiter*' is 'Comma'. The 'File Columns' tab is active, showing a table with the following columns:

Seq*	Column Name*	Reqd	Datatype*	Char Len
1	SNOW_ROUTE	<input type="checkbox"/>	VARCHAR2	30
2	MEMBER_DATUM	<input type="checkbox"/>	VARCHAR2	30
3	START_OFFSET	<input type="checkbox"/>	NUMBER	
4	END_OFFSET	<input type="checkbox"/>	NUMBER	
5	CARDINALITY	<input type="checkbox"/>	NUMBER	
6	MEMBER_SEQUENCE	<input type="checkbox"/>	NUMBER	

Figure 44 shows an example where member elements are being loaded into a Linear Group Type. The Group Type is also 'partial' so the start and end offsets are included in the load file (if the Group was non Partial the start and end offsets may be derived as described in the next section) along with the Cardinality indicator and member sequence number. The membership Start Date and other required information. are either added as constant values or derived using function calls in the Source Column of the 'File Destinations' panel as described in the next section. After the data has been loaded the Parent Routes should be Rescaled (see the Network Manager User Guide).

File Column Settings

The File Column values and the subsequent Holding Table created for loading Group Members is largely determined by the Type of Group into which the member Elements are being loaded. For example when loading member elements into a non partial group the Start and End (BEGIN_MP and END_MP) values are not required as these may be 'derived' using function calls etc, in the Source Column.

File Destinations

The following Columns within the **NM_MEMBERS_ALL** Destination Table should be 'mapped' as described below. Columns not included in this description, e.g. NM_DATE_CREATED are automatically populated on insertion of a record to the table.

NM_NE_ID_IN (Required)

This is the internal unique id of the 'parent' group. Since this value is unlikely to be known it is more common for the Parent Group NE_UNIQUE to be supplied for each record within the data file to be loaded. The Group NE_UNIQUE may be 'decoded' to the internal unique id (ne_id) using a procedure which should be entered in the Source Column for the NM_NE_ID_IN Destination Column. An example is shown below.

```
nm3get.get_ne(pi_ne_unique=>flood_area_membs.flood_area,
pi_ne_nt_type=>
'AREA'). ne_id
```

where **flood_area_membs** is the Unique Ref (Figure 45) of the current File Definition Table, **flood_area** is the Column Name in the Holding Table which stores the value and **AREA** is the Network Type of the Parent Group.

Figure 46 shows an example of this in use with the complete syntax displayed in the Editor.

Figure 45

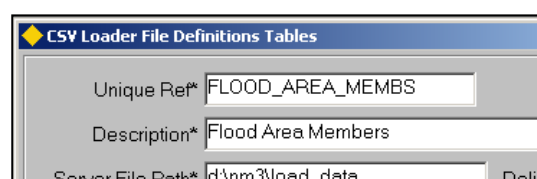
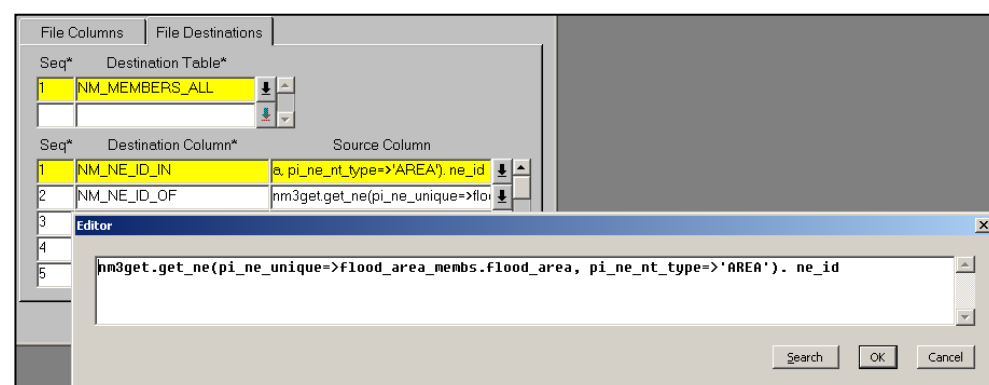


Figure 46



NM_NE_ID_OF (Required)

This is the internal unique id of the member Element (Datum Element, Distance Break or Group). Since this value is unlikely to be known it is more common for the Member element NE_UNIQUE to be supplied for each record within the data file to be loaded. The member NE_UNIQUE may be 'decoded' to the internal unique id (ne_id) using a procedure which should be entered in the Source Column for the NM_NE_ID_OF Destination Column. An example is shown below.

```
nm3get.get_ne(pi_ne_unique=>flood_area_membs.member_rec,
pi_ne_nt_type
=>'ROAD'). ne_id
```

where **flood_area_membs** is the Unique Ref (Figure 45) of the current File Definition Table, **member** is the Column Name in the Holding Table which stores the value and **ROAD** is the Network Type of the member Element.

NM_TYPE (Required)

This is the internal Network Type indicator of the member record and would not generally be included in the load file. A constant value may be entered into the Source Column for this field. The value should be enclosed in single quotes. The NM_TYPE value for group member records should always be set to 'G'.

Figure 47 shows an example where Groups member records are being loaded. The NM_TYPE Destination Column has been set to a constant value of 'G'.

Figure 47

Seq* Destination Table*		
1	NM_MEMBERS_ALL	
Seq* Destination Column* Source Column		
1	NM_NE_ID_IN	nm3get.get_ne(pi_ne_unique=>flood_area_membs.member_rec,
2	NM_NE_ID_OF	nm3get.get_ne(pi_ne_unique=>flood_area_membs.member_rec,
3	NM_TYPE	'G'
4	NM_OBJ_TYPE	'FLOD'

NM_OBJ_TYPE (Required)

This is the Group Type of the parent Group. If this is held within the load file the NM_OBJ_TYPE Column should be 'mapped' to the appropriate column within the Holding Table. Alternatively the Group Type may be entered as a constant value into the Source field. The value should be enclosed in single quotes. Figure 48 shows an example where member Elements are being loaded into a Groups of Type of 'FLOD'.

Figure 48

File Columns File Destinations		
Seq* Destination Table*		
1	NM_MEMBERS_ALL	
Seq* Destination Column* Source Column		
3	NM_TYPE	'G'
4	NM_OBJ_TYPE	'FLOD'
5	NM_BEGIN_MP	'0'

NM_BEGIN_MP **(Required)**

This is the start offset of the member element relative to the parent Group. If the Element is wholly contained within the Group this value will be '0'. However if the Group Type was a Partial Group the Start offset may be a value other than zero if an the element was only partailly contained within the Group. If loading Groups as members of other Groups this value will be '0'.

If the start offset value is held within the load file the NM_BEGIN_MP Column should be 'mapped' to the appropriate column within the Holding Table. Alternatively if the start offset will always be '0' for this load file a constant value may be entered as the Source. The value should be enclosed in single quotes.

NM_START_DATE **(Required)**

This is the start date of the membership record within the Group. If the start date value is held within the load file the NM_START_DATE Column should be 'mapped' to the appropriate column within the Holding Table. If the start date is not held within the load table a constant may be added into the source column, e.g. '01-JAN-1980' (note that the Date must be enclosed in single quotes). This will result in all member records within the load file having the same start date.

A function call to an internal procedure may also be entered as the Source which will 'derive' the start date. For example to generate the Start Date as the current effective date for each of the Nodes being loaded the following function call may be used **NM3USER.GET_EFFECTIVE_DATE**. To set the start date of the member records to the same start date as the parent Group use the function call below.

```
nm3get.get_ne(pi_ne_unique=>flood_area_membs.flood_area,
pi_ne_nt_type=>'AREA'). ne_start_date
```

where **flood_area_membs** is the Unique Ref of the current File Definition Table, **flood_area** is the Column Name in the Holding Table which stores the unique name of the parent and **AREA** is the Network Type of the parent Group.

NM_END_DATE

If loading historic Group membership records enter the source of the End Date. Refer to the previous Column (NM_BEGIN_MP) for options when 'sourcing' the date.

NM_END_MP

This is the end offset of the member element relative to the parent Group. If the Element is wholly contained within the Group this value will be equal to the length of the Element. However if the Group Type was a Partial Group the End offset may be a value other than the total length if an the element was only partailly contained within the Group. If loading Groups as members of other Groups this value will be '0'.

If the end offset value is held within the load file the NM_END_MP Column should be 'mapped' to the appropriate column within the Holding Table. If

loading Groups as members of other Groups a constant value of '0' should be entered as the Source.

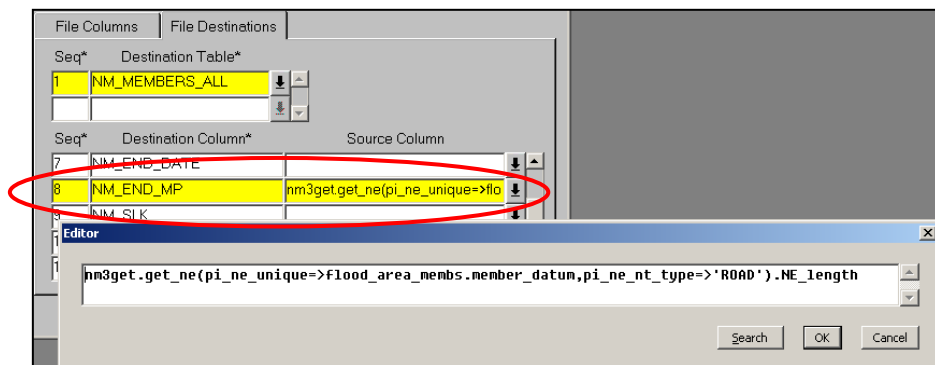
If all elements within the load file are entirely contained within a Group the NM_END_MP may be 'derived' using a function call as shown below.

```
nm3get.get_ne(pi_ne_unique=>flood_area_membs.member_datum,
pi_ne_nt_type=>'ROAD').ne_length
```

where **flood_area_membs** is the Unique Ref of the current File Definition Table, **member_datum** is the Column Name in the Holding Table which stores the unique name of the member record and **ROAD** is the Network Type of the member Element..

Figure 49 shows a working example of this.

Figure 49



NM_SLK (Linear Groups Only)

This is the start Offset of the member element relative to the Route. This value would not normally be loaded and may be calculated once the data has been loaded by Rescaling the Parent Route within the **Groups of Section - NM0110** module (refer to the Network Manager User Guide for information on rescaling Routes).

NM_CARDINALITY (Linear Groups Only)

This is the cardinality Flag of the member Element. If the member Element is in the same cardinal direction as the Parent Group the flag would have a value of '1'. If however the element was in the non cardinal or counter direction the flag would set to '-1'. If all member elements within the load file or of a constant cardinality a the appropriate value could be set as a constant within the Source Column. Alternatively if the Cardinality Flag is being loaded for each member element the Destination Column should be mapped to the appropriate Column within the Holding Table.

NM_ADMIN_UNIT

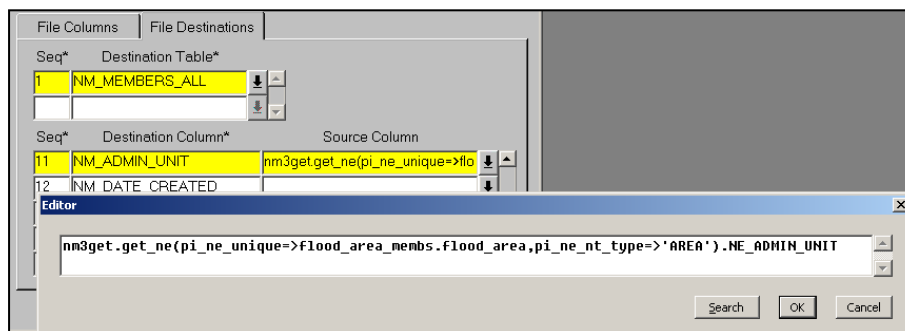
This is the Admin Unit of the Parent Group as defined in the NM_NE_ID_IN Column. This would not often be included in the load file and may be derived using a function call as shown below

```
nm3get.get_ne(pi_ne_unique=>flood_area_membs.flood_area,pi_ne_nt_type
=>
'AREA').NE_ADMIN_UNIT
```

where **flood_area_membs** is the Unique Ref of the current File Definition Table, **flood_area** is the Column Name in the Holding Table which stores the unique name of the Parent record and **AREA** is the Network Type of the Parent Group..

Figure 50 shows a working example of this.

Figure 50



NM_SEQ_NO (Linear Groups Only)

This is the sequence number of the member element. This is the relative position within the Group and hence allows offsets to be calculated. If the sequence number is not included in the load file the member elements will be sequenced in the order in which they are loaded when a Resequence or Rescale operation is conducted on the Parent Route within the **Groups of Section - NM0110** module (refer to the Network Manager User Guide for information on rescaling Routes).

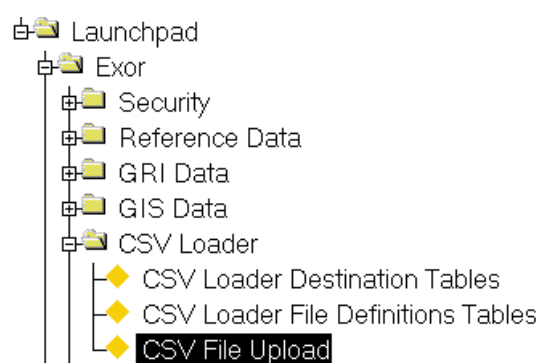
This page intentionally blank

CHAPTER

2

Loading Data

Figure 51
CSV Loader
Menu Option

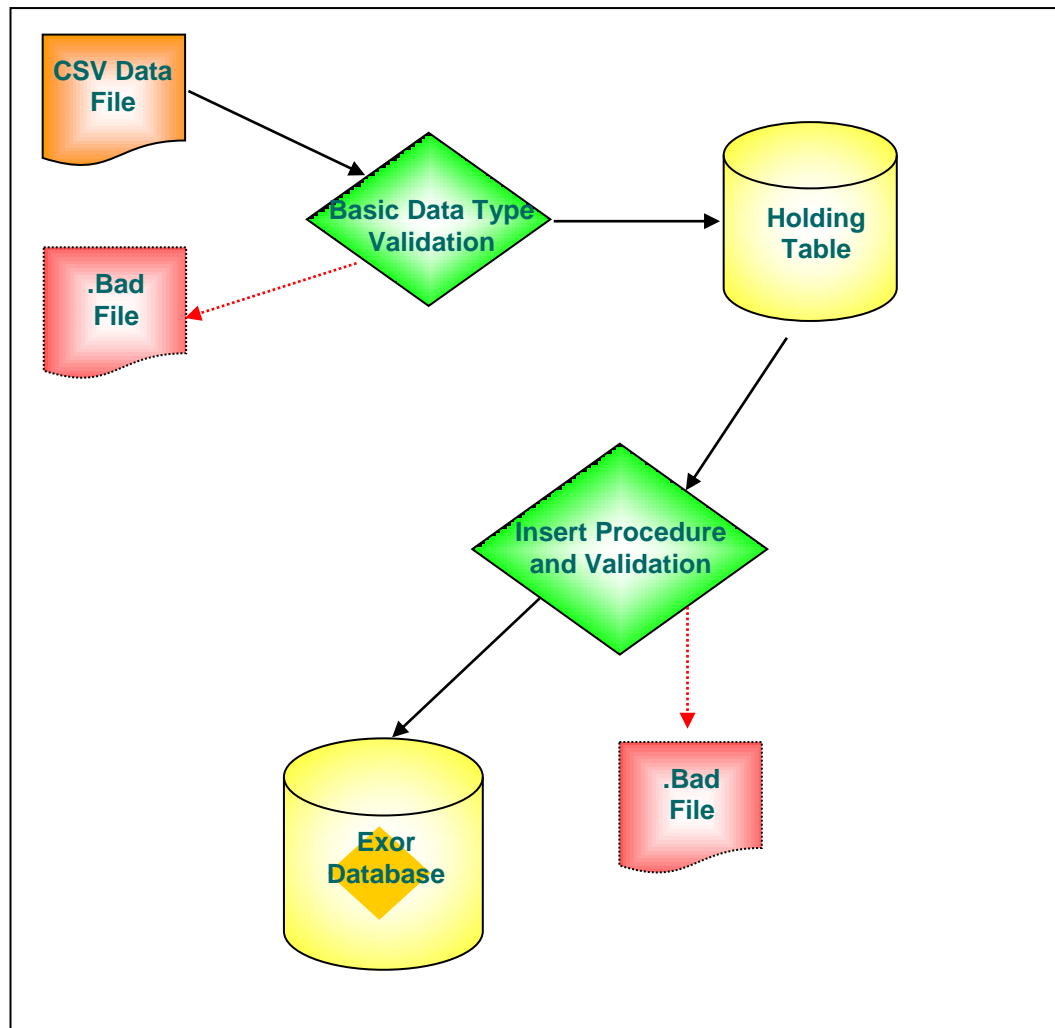


This Chapter describes the process and modules used to load data using the CSV Data loader. The following module is described:

- Load/Process Data - HIGWEB2030

Load/Process Data - HIGWEB2030

Figure 52



CSV File Upload Process

The Loading process is essentially a 2 stage operation. The first stage loads the 'Raw Data' into a Holding Table which is created from the **CSV Loader Files Definitions Tables - HIG2020** module (page 14).

During this first stage of the load into the holding table a basic data type validation is carried out on the CSV data being loaded. This validation checks for the following Data Errors:

Data Type of value being loaded does not match the Data Type specified for the field in the Definition Table

The String length of a field defined as a VARCHAR2 is longer than has been specified for the field in the Definition Table.

A NULL value is present for a field flagged as 'Required' in the Definition Table.

If any errors are detected these are reported on screen with a description of the error and the record number of the erroneous row. This information is also included in a LOG file which gets created during the initial load of data into the Holding Table. Any erroneous Records are written to a .BAD file. If these records are still to be processed the errors may be corrected in the .BAD file and this submitted as a new Load Batch. Alternatively the erroneous records may be corrected in the original CSV data file and the entire file re-submitted as a new Batch. The onscreen display is included in a LOG file which

The naming convention for the .LOG and .BAD files is as follows :

<<Data File Name>>.<<Batch Number>>.LOG (or .BAD)

The Batch number is automatically generated from a sequence and increments by a value of 1 for each new Load Batch. For example, a file named TNC1_LINE.TXT may have a .LOG or .BAD file named :

TNC1_LINE.TXT.381.LOG

Both the .LOG and .BAD files will be created in the directory specified in the 'Server File Path' field of the **CSV Loader Files Definitions Tables - HIG2020** module (page 14).

The second stage of the load process transfers the data which has passed the initial Data Type validation from the holding table to the appropriate Table(s) within the **Exor** Schema. The Destination Table(s) are associated with a File Definition using the File Destinations panel of the **CSV Loader Files Definitions Tables - HIG2020** module.

During the transfer procedure the Data is validated for accuracy against the appropriate **Highways** Metadata. For example, in the case of an Asset Load the validation would include the following :

- Max and Min Values are within defined ranges
- Domain Lookup values exist
- The location at which the Item is being loaded exists, if appropriate
- The location has not already been assigned to another Admin Unit
- No Cross Attribute or Cross Item validation Rules have been broken.

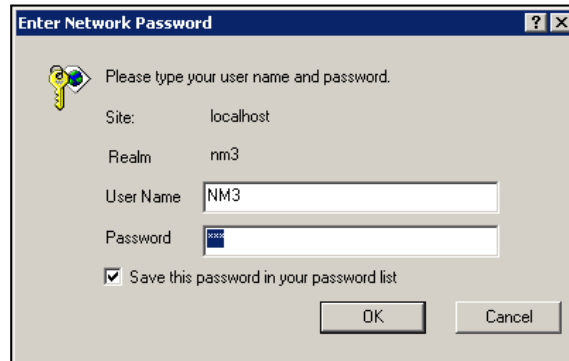
The validation which is carried out on Data during this stage is determined by Destination Table(s) and the Validation Procedure specified for the Destination Table in the **CSV Loader Destination Tables - HIG2010** module.

If any errors are detected these are reported on screen with a description of the error and the record number of the erroneous row in the same fashion as the 'stage 1' load into the holding table. Erroneous data is written to the .BAD file for the Load Batch along with any Data Errors that were detected during the initial load into the Holding Table.

Note that the presence of erroneous data will not prevent any valid data from being successfully transferred.

Data within a Holding Table may be validated without actually transferring it to its Destination Table(s).

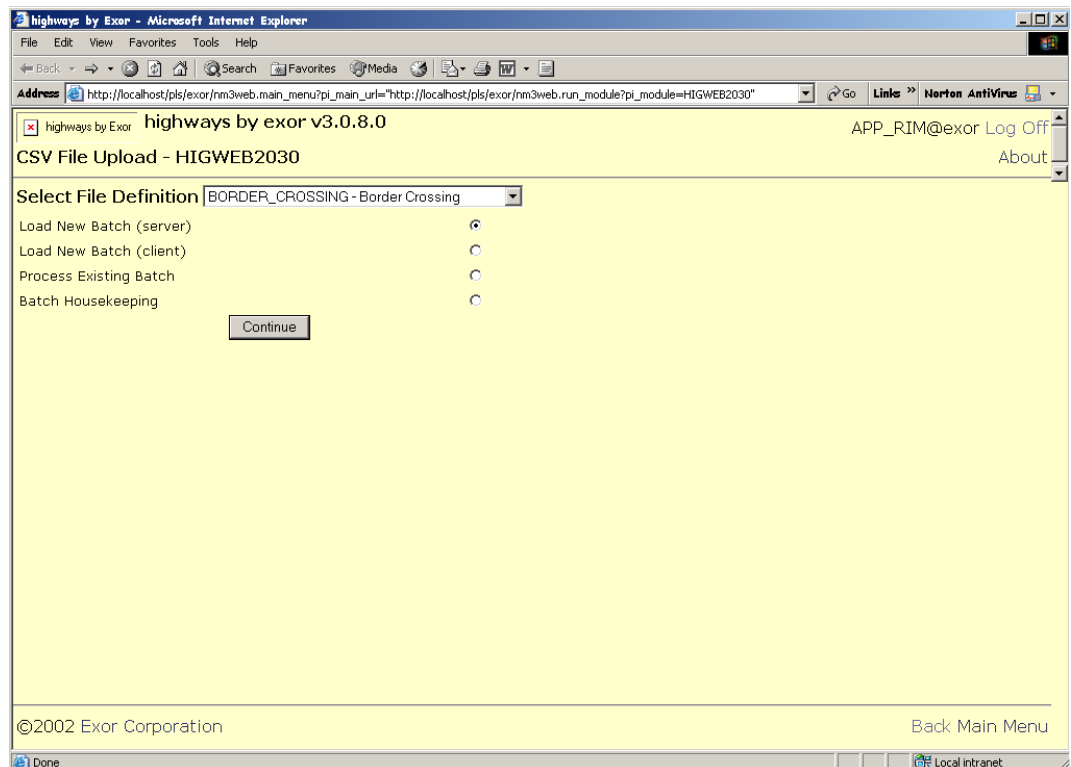
Figure 53



Logging onto a Web Session

When either the **Web Main Menu - NMWEB0000** or the **Load/Process Data - HIGWEB2030** option is opened your Web Browser will be called and a Dialogue displayed to allow the User to enter their Username and Password. Once this is done and the **[OK]** button is pressed the first CSV File Upload page will be displayed as shown in Figure 54.

Figure 54



Hyper Links

A number of Links are included on each Web Page. These are namely:

Log Off (Hyperlink)

Logs the User out of the current Web Session.

About (Hyperlink)

Displays the Current Version of all Exor Products installed

Exor Corporation (Hyperlink)

Calls <http://www.exorcorp.com/>

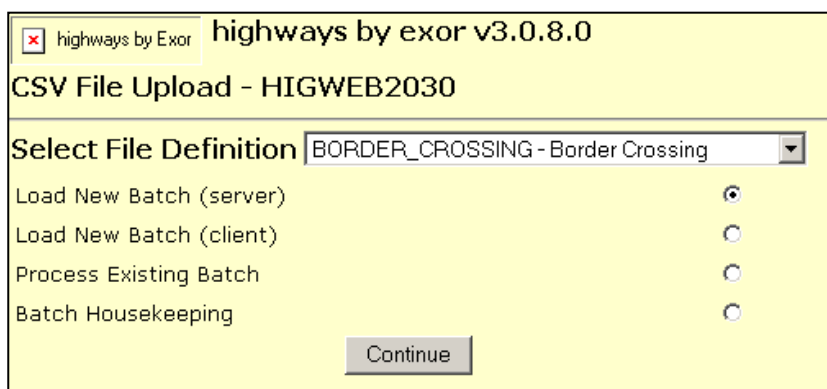
Back (Hyperlink)

Goes back to previous page

Main Menu (Hyperlink)

Calls the Web Main Menu - NMWEB0000

Figure 55



Selecting the File Definition

The first step in the CSV File Upload is to select the File Definition Table for the Load Batch. These will have been previously defined in the **CSV Loader File Definitions Table - HIG2020** module (page 14).

Select File Definition (Required) List

Select the required File Definition. The List of Values will display the File Definition Unique Reference and Description.

Load New Batch (Server) (Radio Button)

Select this option if a new Load Batch is to be created, i.e. a CSV data file is to be processed and the data file is located on the Server Machine

Load New Batch (Client) (Radio Button)

Select this option if a new Load Batch is to be created, i.e. a CSV data file is to be processed and the data file is located on the Client PC.

Process Existing Batch (Radio Button)

Select this option to process an existing Batch. This will allow a Batch that has already been loaded into a Holding Table to be processed and

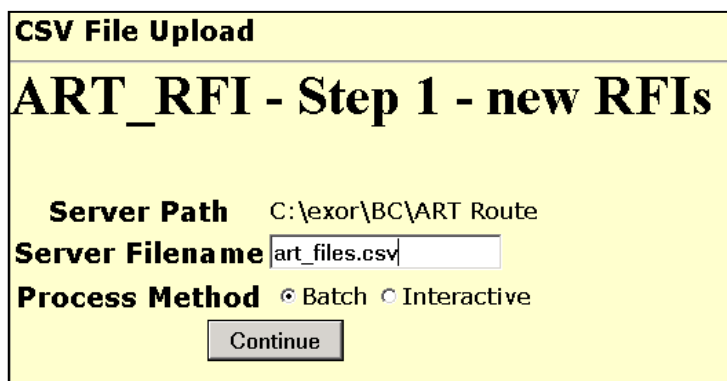
'transferred' to the appropriate Destination Table within the Highways Schema.

Batch Houskeeping (Radio Button)

Select this option if you wish to remove existing Load Batches which have

To continue the Load Process press the **[Continue]** button in the Web Page.

Figure 56



CSV File Upload

ART_RFI - Step 1 - new RFIs

Server Path C:\exor\BC\ART Route

Server Filename

Process Method ☒ Batch ☐ Interactive

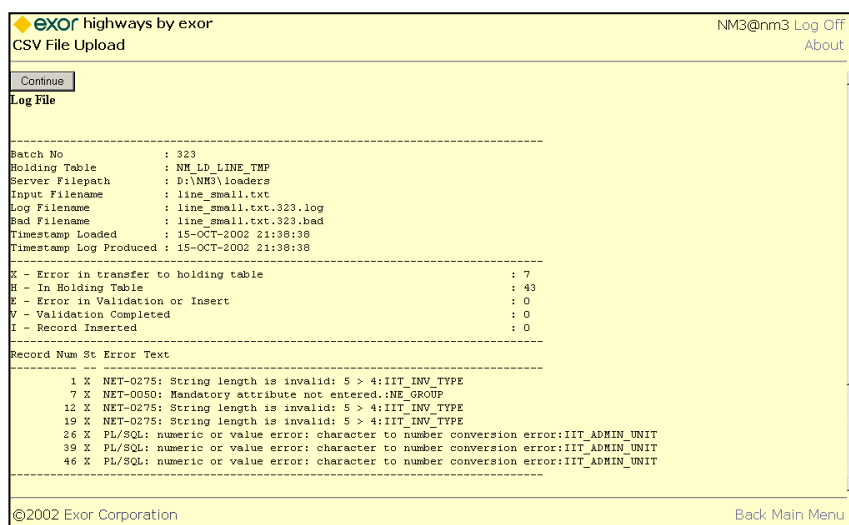
Load New Batch from Server

If the 'Load New Batch (Server)' option is selected a new Web Page will be displayed to allow the Filename of required CSV data file to be entered. The Unique Reference and Description of the selected File Definition Table will be displayed along with the Server Path where the Load file must be located. The Server Path is defined for the Definition Table in the **CSV Loader Files Definitions Tables - HIG2020** module (page 14).

Enter the required Filename. Select a process method either Batch or Interactive. Selecting 'Batch' will initiate the load and then send an email when the load is complete. Selecting 'Interactive' will initiate the load and display results when the load is complete. If you select 'Batch' you should ensure that NM3Mail is correctly set up and that your user account has an email address associated with it. It can be useful to run in Batch Mode if you have a large number of records to process. To begin the Data Load process, press the **[Continue]** button. The data contained within the CSV Data File will be validated and loaded into a Holding Table as previously described. A LOG file and a BAD file will be created. The LOG file will contain a summary of the Loaded data along with any Error messages which occurred during the data validation. Erroneous records are written to the bad file.

Figure 57 shows an example of the Log file generated during the first stage of the Load Process.

Figure 57



EXOR highways by exor

CSV File Upload

NM3@nm3 Log Off
About

Log File

```

-----
Batch No           : 323
Holding Table      : NM_LD_LINE_THP
Server Filepath    : D:\NM3\loaders
Input Filename     : line_small.txt
Log Filename       : line_small.txt.323.log
Bad Filename       : line_small.txt.323.bad
Timestamp Loaded   : 15-OCT-2002 21:38:38
Timestamp Log Produced : 15-OCT-2002 21:38:38
-----
X - Error in transfer to holding table           : 7
H - In Holding Table                           : 43
E - Error in Validation or Insert                : 0
V - Validation Completed                        : 0
I - Record Inserted                             : 0
-----
Record Num St Error Text
-----
1 X NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
7 X NET-0050: Mandatory attribute not entered:INE_GROUP
12 X NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
19 X NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
26 X PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
39 X PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
46 X PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
-----

```

©2002 Exor Corporation

Back Main Menu

The LOG File contains the following information:

- Batch No
- Holding Table Name
- Server Filepath
- Input Filename
- Log Filename
- Bad Filename
- Timestamp Loaded
- Timestamp Log Produced

A summary of the Number of Records at each Status. The Status Codes are as follows:

X	-	Error in transfer to holding table
H	-	In Holding Table
E	-	Error in Validation or Insert
V	-	Validation Completed
I	-	Record Inserted

If any errors are detected during the data validation an error message will be displayed for each erroneous record. The Record Number and Status are included for each record. The Erroneous records are written to the BAD file. The table below shows some examples of typical errors which may occur.

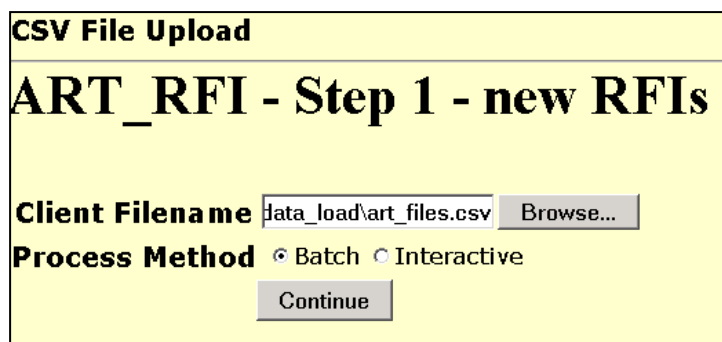
Record No	Status	Error Message
1	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
7	X	NET-0050: Mandatory attribute not entered.:NE_GROUP
12	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
19	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
26	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
39	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
46	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT

Erroneous records may be corrected in the BAD file and resubmitted as a new Load Batch upon completion of the current Batch or the original CSV data file corrected and resubmitted without further processing the current Batch.

To continue the current Load Batch press the **[Continue]** button at the top of the Web Page. If the original CSV data file is to be rechecked and resubmitted without finishing the current Batch press the **[Back]** hyper link, correct the data file and press **[Continue]** to reload.

Refer to page 65 for information on the second stage of the load process.

Figure 58



CSV File Upload

ART_RFI - Step 1 - new RFIs

Client Filename data_load\art_files.csv **Browse...**

Process Method ☒ Batch ☐ Interactive

Continue

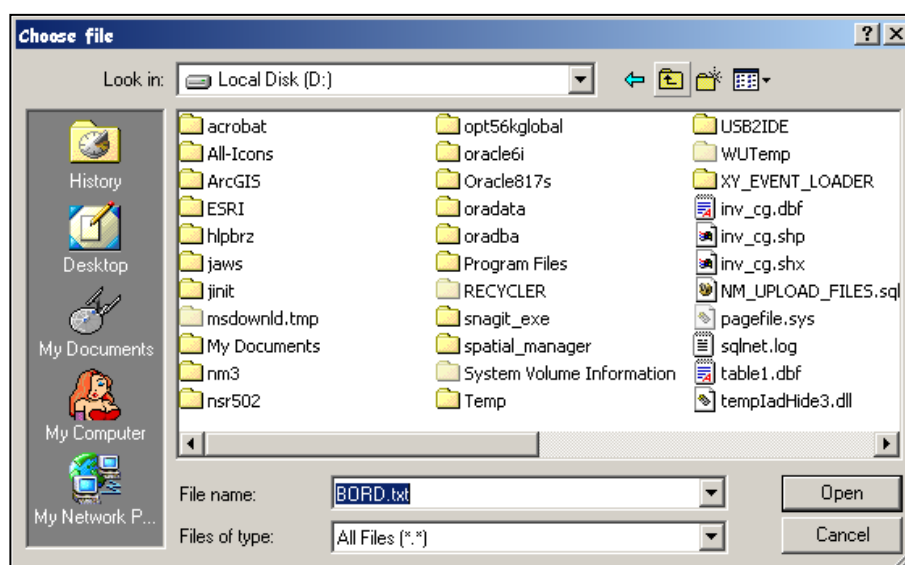
Load New Batch from Client

If the 'Load New Batch (Client)' option is selected a new Web Page will be displayed to allow the Path and Filename of required CSV data file to be entered.

Note that this must be the full path and filename of the load file on the Client PC, e.g. c:\exor\load_files\bord.txt.

To search for the load file on the Client PC press the **[Browse]** button. This will call a standard windows 'Choose File' dialogue as shown in Figure 59.

Figure 59



Once the full path and filename have been entered select a process method either Batch or Interactive. Selecting 'Batch' will initiate the load and then send an email when the load is complete. Selecting 'Interactive' will initiate the load and display results when the load is complete. If you select 'Batch' you should ensure that NM3Mail is correctly set up and that your user account has an email address associated with it. It can be useful to run in Batch Mode if you have a large number of records to process. To start the load press the **[Continue]** button.

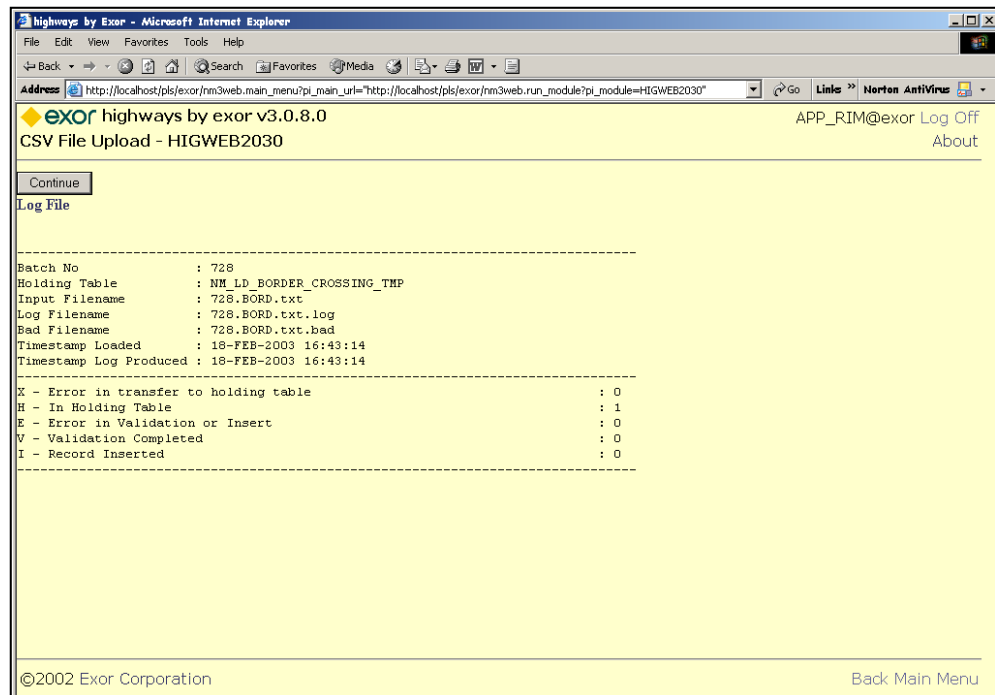
The data file will be loaded into a database table (nm_upload_files) from where it will be processed, validated and loaded into the Temporary Holding table as previously described.

The name of the file within the nm_upload_files table will be appended with the batch number (this is generated from a database sequence), e.g. if the load file is called CULV.TXT it may be called 719.CULV.TXT in the nm_upload_files table, where 719 is the batch number.

A LOG file and a BAD file will be created in the nm_upload_files table. The LOG file will contain a summary of the Loaded data along with any Error messages which occurred during the data validation. Erroneous records are written to the .bad file.

Figure 60 shows an example of the log file created during the first stage load process.

Figure 60



The LOG File contains the following information:

- Batch No
- Holding Table Name
- Input Filename (with appended batch number)
- Log Filename
- Bad Filename
- Timestamp Loaded
- Timestamp Log Produced

A summary of the Number of Records at each Status. The Status Codes are as follows:

- | | | |
|----------|---|------------------------------------|
| X | - | Error in transfer to holding table |
| H | - | In Holding Table |
| E | - | Error in Validation or Insert |
| V | - | Validation Completed |
| I | - | Record Inserted |

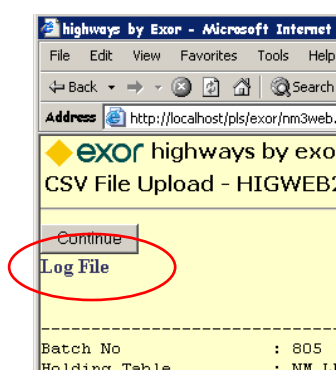
If any errors are detected during the data validation an error message will be displayed for each erroneous record. The Record Number and Status are included for each record. The Erroneous records are written to the BAD file. The table below shows some examples of typical errors which may occur.

Record No	Status	Error Message
1	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
7	X	NET-0050: Mandatory attribute not entered.:NE_GROUP
12	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
19	X	NET-0275: String length is invalid: 5 > 4:IIT_INV_TYPE
26	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
39	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT
46	X	PL/SQL: numeric or value error: character to number conversion error:IIT_ADMIN_UNIT

Erroneous records may be corrected in the .BAD file and resubmitted as a new Load Batch upon completion of the current Batch or the original CSV data file corrected and resubmitted without further processing the current Batch.

To save the Log or Bad file to the Client PC Click on the word 'Log File' or 'Bad File' on the window (Figure 61). This is a hyper Link to the file.

Figure 61



A standard windows File Download Dialogue will be displayed allowing the User to open the file or save to the Client PC.

To continue the current Load Batch press the **[Continue]** button at the top of the Web Page. If the original CSV data file is to be corrected and resubmitted without finishing the current Batch press the **[Back]** hyper link, correct the data file and press **[Continue]** to reload.

Refer to page 65 for information on the second stage of the load process.

Figure 62

CSV File Upload

ART_RFI - Step 1 - new RFIs

Select Batch

1642 - "art_rfi.csv" (12-DEC-2003 09:34:25)

Process

Load

Produce Log Files

Process Method

☒ Batch

☐ Interactive

Continue

Load / Validate Data (2nd Stage)

Once the data has been successfully loaded into a Holding table it may be Validated for data accuracy against the metadata defined for the Object type within the **Exor** database without actually loading it into the appropriate Destination Tables. For example, if loading Asset Items, validation will be carried out to ensure that Max and Min Values are within defined ranges or Domain Lookup values exist for example. When the Validation is completed the LOG and BAD files will be updated accordingly.

Select Batch	(Required)	List
Select the Batch to process through the second stage of the CSV File Upload. The Batch Name will include the Data File Name along with the Date and Time of the stage load.		

Validate	(Radio Button)
Select this option is data is to be validated but not transferred to the appropriate Destination Table.	

Load	(Radio Button)
Select this option to Transfer the data from a Holding Table to the appropriate Destination Table. During this transfer the data is validated and erroneous data is written to the BAD file. The LOG file for the Load Batch will be updated to reflect the Number of Records loaded, etc as well as any Error messages which result from the data validation. The LOG file will be displayed on screen for review.	

Produce Log Files	(Radio Button)
Select this option to recreate the log files for the currently selected batch.	

Once the batch has been chosen select a process method either Batch or Interactive. Selecting 'Batch' will initiate the load and then send an email when the load is complete. Selecting 'Interactive' will initiate the load and display results when the load is complete. If you select 'Batch' you should ensure that NM3Mail is correctly set up and that your user account has an email address associated with it. It can be useful to run in Batch Mode if you have a large number of records to process. To start the load press the **[Continue]** button.

To load another Data File press the **[Continue]** button.

Figure 63

highways by Exor - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Back Forward Stop Home Search Favorites Media

Address http://localhost/pls/exor/nm3web.main_menu?pi_main_url=http://localhost/pls/exor/nm3web.run_module?pi_module=HIGWEB2030

exor highways by exor v3.0.8.0

APP_RIM

CSV File Upload - HIGWEB2030

BORDER_CROSSING - Border Crossing

Batch Housekeeping

Batch No	Filename	Source	Rec Cnt	Batch Dets	Created	Created By	Last Modified	Modified By	Delete				
628	BORDERS.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>E</td><td>1</td></tr></table>	Status	Count	E	1	26-NOV-2002 13:02:27	APP_RIM	16-FEB-2003 13:33:05	APP_RIM	<input type="checkbox"/>
Status	Count												
E	1												
629	BORDERS.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>I</td><td>1</td></tr></table>	Status	Count	I	1	26-NOV-2002 13:07:45	APP_RIM	16-FEB-2003 13:33:05	APP_RIM	<input type="checkbox"/>
Status	Count												
I	1												
631	BORDERS.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>I</td><td>1</td></tr></table>	Status	Count	I	1	26-NOV-2002 13:20:00	APP_RIM	16-FEB-2003 13:33:05	APP_RIM	<input type="checkbox"/>
Status	Count												
I	1												
716	BORD.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>X</td><td>1</td></tr></table>	Status	Count	X	1	17-FEB-2003 22:17:47	APP_RIM	17-FEB-2003 22:17:47	APP_RIM	<input type="checkbox"/>
Status	Count												
X	1												
717	BORD.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>H</td><td>1</td></tr></table>	Status	Count	H	1	17-FEB-2003 22:18:15	APP_RIM	17-FEB-2003 22:18:15	APP_RIM	<input type="checkbox"/>
Status	Count												
H	1												
718	BORD.TXT	S	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>X</td><td>1</td></tr></table>	Status	Count	X	1	17-FEB-2003 22:18:38	APP_RIM	17-FEB-2003 22:18:38	APP_RIM	<input type="checkbox"/>
Status	Count												
X	1												
719	719.BORD.txt 719.BORD.txt.log 719.BORD.txt.bad	C	1	<table><tr><th>Status</th><th>Count</th></tr><tr><td>X</td><td>1</td></tr></table>	Status	Count	X	1	17-FEB-2003 22:57:46	APP_RIM	17-FEB-2003 22:57:47	APP_RIM	<input type="checkbox"/>
Status	Count												
X	1												

Batch Housekeeping

Previously loaded batches may be removed from the system using the Batch Housekeeping option on the CSV File Upload - HIGWEB2030 Web Page. This will remove all records associated with the load batch from all the appropriate **Highways** Tables.

The Batch Housekeeping option will display the details for each of the load batches for the selected File Definition. The details displayed are :

- Batch Number
- Filename (including Log and Bad file names if loaded from Client PC)
- Source (either S- Server or C- Client)
- Record Count (number of records in the batch)
- The Status Codes and Record count for each applicable status within the Batch
- Date and Time Created
- Created By
- Last Modified Date and Time
- Modified By

Select the 'Delete' Check box for the batches which are to be removed. Once all the required Batches have been selected press the **[Delete]** button on the page.

Index

B

Batch Housekeeping · 70

C

CSV File Upload Process · 58

CSV Loader Destination Table · 6, 8, 20, 23

CSV Loader File Definitions Table · 5, 9, **10, 11, 13, 14**, 15, 17, 18, 19, 62

CSV Loader Files Definitions Tables - HIG2020 · 6, 7, 8, 58, 59, 64

D

Default Values Panel · 8

Destination Table · 6, 7, 8, 9, **10, 11, 13, 14**, 15, 17, 18, 19, 23, 24, 25, 27, 28, 29, 30, 35, 36, 37, 41, 42, 43, 45, 46, 48, 50, 51, 59, 60, 63, 69

Destination Tables · 5, 6, 7, 9, **10, 11, 13**, 15, 17, 18, 20, 23, 25, 27, 33, 35, 39, 40, 41, 59, 69

F

File Column Settings · 26, 34, 40, 51

File Columns · 21, 22, 23, 26, 34, 40, 44

File Columns Panel · 21

File Definitions Table · 20

File Destination Settings · 27, 35, 41

File Destinations · 6, 8, 23, 45, 50, 51, 59

File Destinations Panel · 23

H

HIGWEB2030 · 57, 58, 60, 70

Holding Table · 5, 7, 19, 20, 21, 22, 23, 26, 27, 28, 29, 30, 34, 35, 36, 37, 38, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 58, 59, 60, 62, 64, 65, 67, 69

I

Insert Procedure · 7, 9, **10, 11, 13**, 15, 17, 18

L

Load / Validate Data (2nd Stage) · 69

Load New Batch from Client · 66

Load New Batch from Server · 64

Load/Process Data · 57, **58, 60**

Loading Asset · 9, 25

Loading Asset Item Data · 9

Loading Asset Item Location Data · 9

Loading Group Members · 50

Loading Network Elements · 18, 43

Loading Network Node Points · 17

Loading Network Nodes · 23, 39

Logging onto a Web Session · 60

N

NM_NODES_ALL · 17, 39, 41, 42, 43

NM_POINTS · 17, 39, 41, 42, 43

NMWEB0000 · 60, 62

R

Reference Post · 3, **14**, 32, 33, 37, 38

Reference Post Offset · 14, 32

Route Mile Post' referencing · 14, 32

S

Selecting the File Definition · 62

Server File Path · 20, 59

V

V_LOAD_INV_MEM_ELE_MP_AMBIG · 9, 10, 13, 14

V_LOAD_INV_MEM_ELE_MP_EXCL · 9, 10, 11

V_LOAD_INV_MEM_ON_ELEMENT · 9, **10**, 15, 27, 29, 35, 37

V_LOAD_LOCATE_INV_BY_REF · 2, 3, **14**, 15, 37

Validation Procedure · 7, 9, **10, 11, 14**, 59