

Exor Corporation Limited

The Global Leader in

Infrastructure Asset Management



Exor V 4.5

CSV Data Loader



This page intentionally blank

#### 

Quality Assurance Statement

|  |  |
| --- | --- |
| Document Details | |
| **File:**  CSV Data Loader v4.5.doc | **Prepared by:**  T.C. Stewart |
| **Manual Name:**  CSV Data Loader | **Reviewed by:** |
| **Version:**  1.0 | **Approved for issue by:** |
| **Date of Issue:**  Dec 2006 | **Product Manager:** |
| **File Name:**  Document Centre/Exor/Product Manuals/4.5.0.0 Product Manuals/4.5.0.0 completed documentation/ CSV Data Loader v4.5.doc | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 |
| 4 | May 2011 | IS | None | No changes for 4.4 |
| 5 | Nov 2011 | IS | None | No changes for v 4.5 |

CONTENTS

[Quality Assurance Statement 2](#_Toc308609046)

[CSV Loader Destination Table - HIG2010 5](#_Toc308609047)

[Destination Tables 6](#_Toc308609048)

[Default Values Panel 7](#_Toc308609049)

[Loading Asset Item Data 8](#_Toc308609050)

[Loading Asset Item Location Data 8](#_Toc308609051)

[V\_LOAD\_INV\_MEM\_ON\_ELEMENT 8](#_Toc308609052)

[V\_LOAD\_INV\_MEM\_ELE\_MP\_EXCL 9](#_Toc308609053)

[V\_LOAD\_INV\_MEM\_ELE\_MP\_AMBIG 11](#_Toc308609054)

[V\_LOAD\_LOCATE\_INV\_BY\_REF 12](#_Toc308609055)

[Loading Network Node Points 14](#_Toc308609056)

[Loading Network Elements (Datum or Groups) 15](#_Toc308609057)

[CSV Loader File Definitions Table - HIG2020 16](#_Toc308609058)

[File Definitions Table 17](#_Toc308609059)

[File Columns Panel 18](#_Toc308609060)

[Creating a Holding Table 19](#_Toc308609061)

[File Destinations Panel 20](#_Toc308609062)

[Loading Asset 22](#_Toc308609063)

[File Column Settings 23](#_Toc308609064)

[File Destination Settings 24](#_Toc308609065)

[NM\_INV\_ITEMS (Asset Item Details) 24](#_Toc308609066)

[V\_LOAD\_INV\_MEM\_ON\_ELEMENT (Asset Item Locations) 26](#_Toc308609067)

[Loading Asset Data using Reference Post Referencing 29](#_Toc308609068)

[File Column Settings 31](#_Toc308609069)

[File Destination Settings 32](#_Toc308609070)

[NM\_INV\_ITEMS (Asset Item Details) 32](#_Toc308609071)

[*V\_LOAD\_LOCATE\_INV\_BY\_REF* (Locate Assets using Reference Post Referencing) 34](#_Toc308609072)

[Loading Network Nodes 36](#_Toc308609073)

[File Column Settings 37](#_Toc308609074)

[File Destination Settings 38](#_Toc308609075)

[NM\_POINTS 38](#_Toc308609076)

[NM\_NODES\_ALL 39](#_Toc308609077)

[Loading Network Elements (Datum or Groups) 40](#_Toc308609078)

[File Columns 41](#_Toc308609079)

[File Destinations 41](#_Toc308609080)

[Loading Group Members 46](#_Toc308609081)

[File Column Settings 47](#_Toc308609082)

[File Destinations 47](#_Toc308609083)

[Load/Process Data - HIGWEB2030 54](#_Toc308609084)

[CSV File Upload Process 54](#_Toc308609085)

[Logging onto a Web Session 56](#_Toc308609086)

[Hyper Links 57](#_Toc308609087)

[Selecting the File Definition 57](#_Toc308609088)

[Load New Batch from Server 59](#_Toc308609089)

[Load New Batch from Client 61](#_Toc308609090)

[Load / Validate Data (2nd Stage) 64](#_Toc308609091)

[Batch Housekeeping 65](#_Toc308609092)

CHAPTER

1

Introduction

Figure

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 CSV

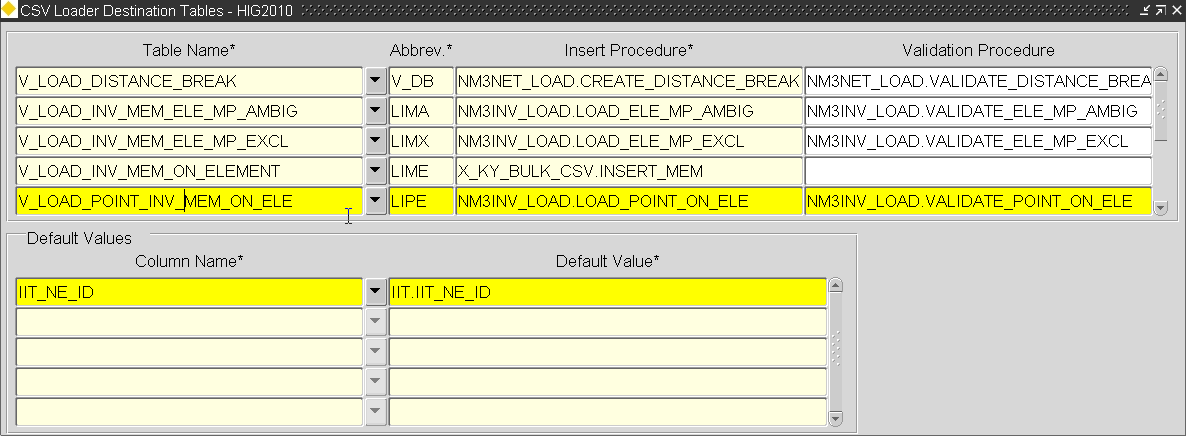
Loader Destination

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

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.

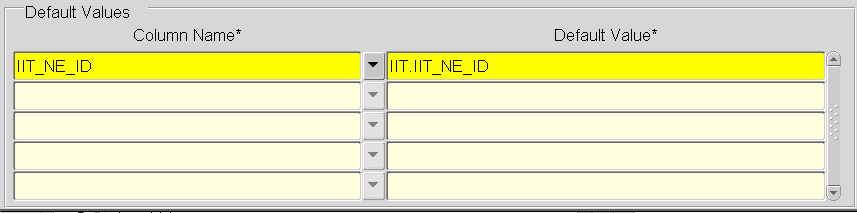
Insert Procedure (Required)

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**.

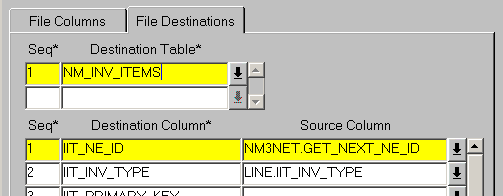
Figure



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>), ‘ ‘)

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

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

|  |  |  |  |
| --- | --- | --- | --- |
| **Table Name** | **Abbrev** | **Insert Procedure** | **Validation Procedure** |
| NM\_INV\_ITEMS | IIT | NM3INS.INS\_IIT\_ALL | NM3INV.VALIDATE\_REC\_IIT |

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

|  |  |
| --- | --- |
| **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.

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

|  |  |
| --- | --- |
| Table Name | V\_LOAD\_INV\_MEM\_ON\_ELEMENT |
| **Abbrev** | LIME |
| **Insert Procedure** | NM3INV\_LOAD.LOAD\_ON\_ELEMENT |
| **Validation Procedure** | NM3INV\_LOAD.VALIDATE\_ON\_ELEMENT |

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

|  |  |
| --- | --- |
| **Column Name** | **Default Value** |
| IIT\_NE\_ID | IIT.IIT\_NE\_ID |

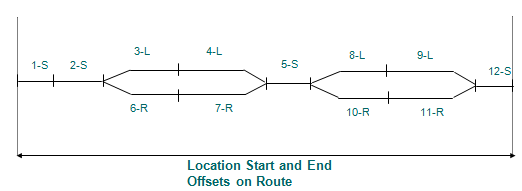
Note that the 'IIT' value which prefixes IIT\_NE\_ID in the Default Value field is the Abbreviation used for the NM\_INV\_ITEMS Destination table.

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



The Route displayed in 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

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

|  |  |
| --- | --- |
| 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 |

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

|  |  |
| --- | --- |
| 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.

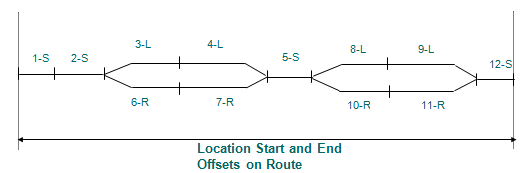
V\_LOAD\_INV\_MEM\_ELE\_MP\_AMBIG

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

Extent limits



The Route displayed in 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

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

|  |  |
| --- | --- |
| 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 |

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

|  |  |
| --- | --- |
| 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

1

2

3

4

Route 1

0.75

0.25

0.1

0.85

1.75

Asset A

3.1

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 Refernce 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

|  |  |
| --- | --- |
| 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

|  |  |
| --- | --- |
| **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

|  |  |  |
| --- | --- | --- |
| 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

|  |  |
| --- | --- |
| **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

|  |  |
| --- | --- |
| 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

|  |  |  |
| --- | --- | --- |
| **Table Name** | **Abbrev** | **Insert Procedure** |
| NM\_ELEMENTS\_ALL | NE | NM3NET.INSERT\_ANY\_ELEMENT |

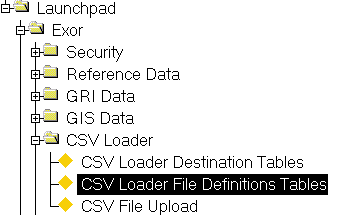
Note that a Validation procedure is not normally required.

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

|  |  |
| --- | --- |
| **Column Name** | **Default Value** |
| NE\_ID | nm3seq.next\_NE\_ID\_SEQ |

CSV Loader File Definitions Table - HIG2020

Figure

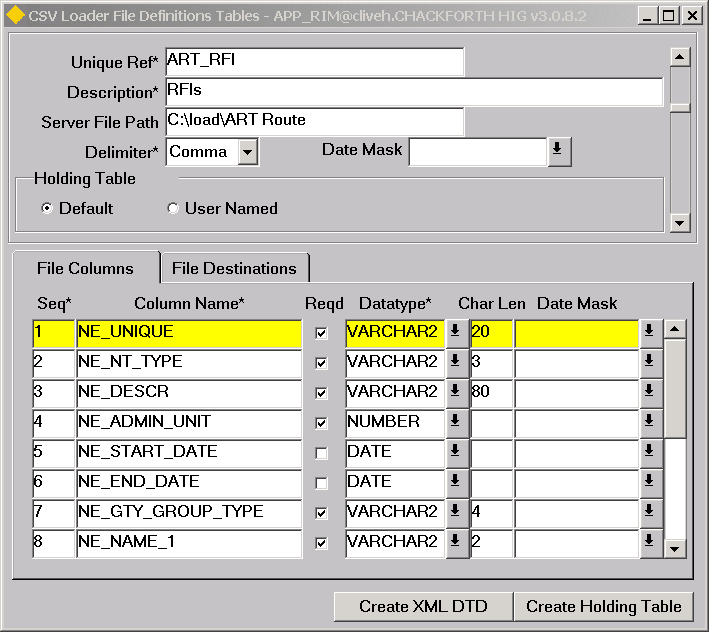
CSV Loader

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

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 maximim 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 61 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.

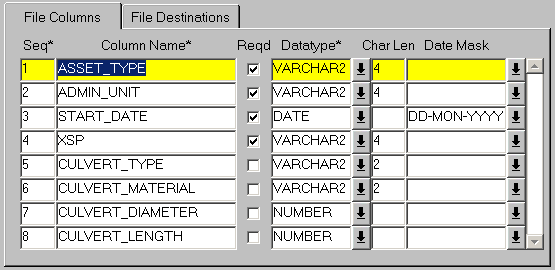
Holding Table (Required) Radio Button

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

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)

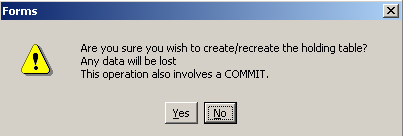
Enter the maximum length of a character string allowed in this Column. If a record has a value which is longer than the specified length it will fail the Basic Data Type validation during the Load into the Holding Table.

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 () 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

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

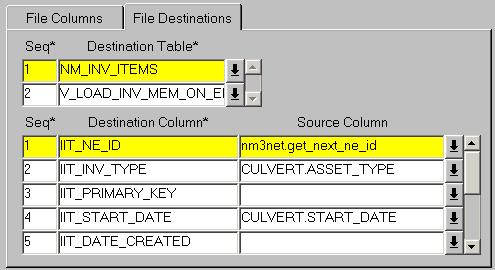
NM\_LD\_<<File Definition Unique>>\_TMP

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

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.

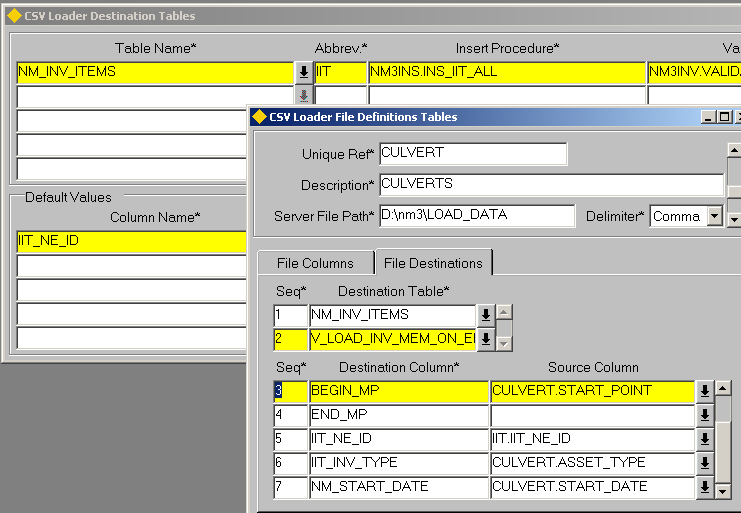
Each of the required Destination Columns must be mapped to their data Source using the 'Source Column' field.

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. 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



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 conatined 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. shows the flexible Attributes have been defined for Culvert Asset Type using the ***Asset Metamodel - NM0410*** module.

Table

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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 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

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

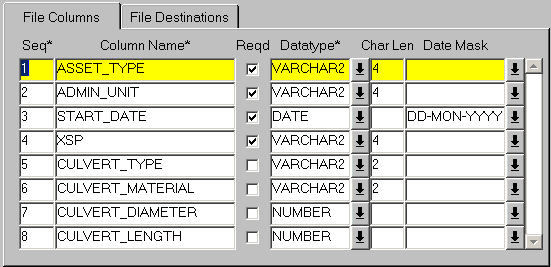
CULV,REG1,01-JAN-2000,LD,S,C,15,12.5,H26,0,5.78

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

|  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **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

File Column Settings

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

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Seq** | **Column Name** | **Datatype** | **Char Len** |
| **1** | **ASSET\_TYPE** | **VARCHAR2** | **4** |
| **2** | **ADMIN\_UNIT** | **VARCHAR2** | **4** |
| **3** | **START\_DATE** | **DATE** |  |
| **4** | **XSP** | **VARCHAR2** | **4** |
| **5** | **CULVERT\_TYPE** | **VARCHAR2** | **2** |
| **6** | **CULVERT\_MATERIAL** | **VARCHAR2** | **2** |
| **7** | **CULVERT\_DIAMETER** | **NUMBER** |  |
| **8** | **CULVERT\_LENGTH** | **NUMBER** |  |
| **9** | **ROUTE** | **VARCHAR2** | **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'. shows an example where an Asset Type of BORD has been defined as the Asset Type.

Figure



START\_DATE

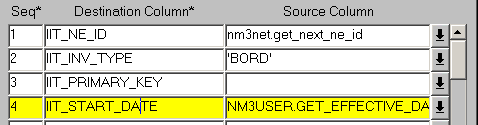
A Start date for both the Asset Item and the Asset Item Location (if located) is required. This value would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

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. shows an example of this in use.

Figure



IIT\_ADMIN\_UNIT

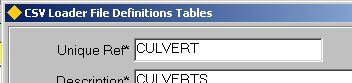
This is the Admin Unit of the Asset Item. The Insert procedures used when transferring the data from the Holding Table to the Destination Table expects the internal unique of the Admin Unit to be supplied. Since this value is unlikely to be known it is more common for the Admin Unit name to be supplied 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=>culvert.admin\_unit,pi\_nau\_admin\_type=>'MOT').nau\_admin\_unit

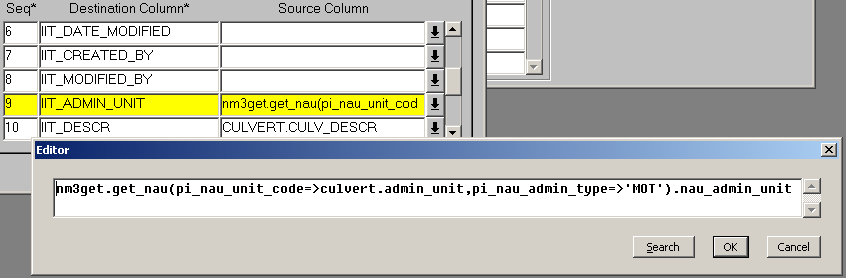
where **culvert** is the Unique Ref () of the current File Definition Table and **MOT** is the Admin Type Code of the Admin Unit

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

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



Figure

Figure

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

|  |  |  |
| --- | --- | --- |
| **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

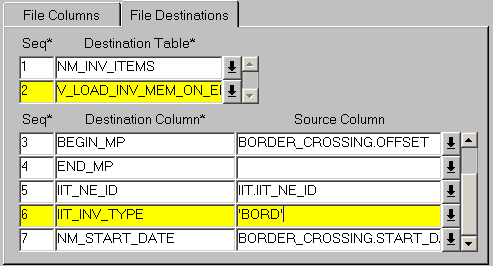
This is the internal unique id of the Asset Item. A default value of ***IIT.IIT\_NE\_ID***  is supplied as standard metadata for the Destination Column (page 7). This is a 'mapping' to the **IIT\_NE\_ID** Column of the NM\_INV\_ITEMS Destination Table which has the Abbrevaition of **IIT.**

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



START\_DATE

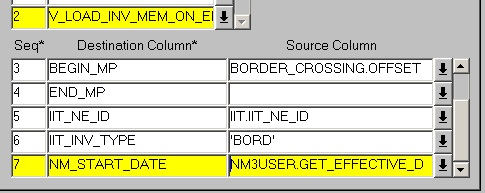
A Start date for both the Asset Item and the Asset Item Location (if located) is required. This value would normally be included within each record of the load file. If this is the case this Column should be mapped to the appropriate Column in the Holding Table.

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



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’, 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 below.

Figure

1

2

3

4

Route 1

0.75

0.25

0.1

0.85

1.75

Asset A

3.1

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 usd 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

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **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

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

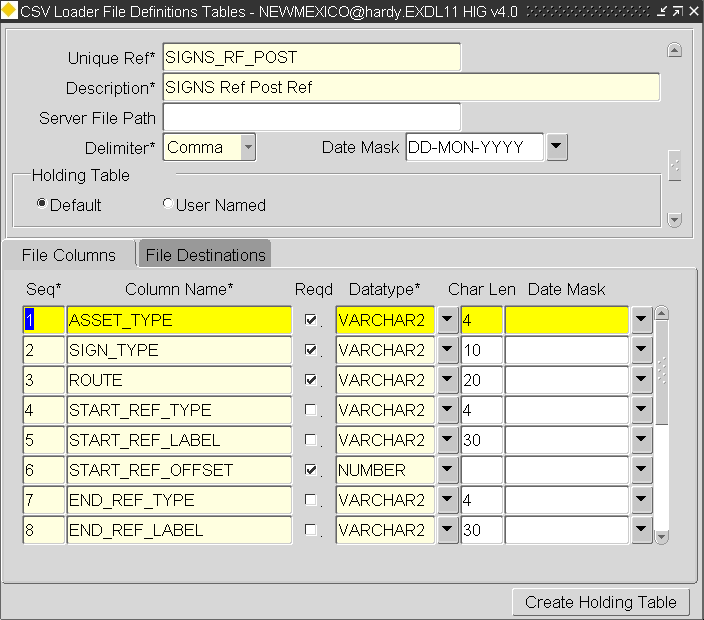
SIGN,STOP,BIA-AIA013-P,MP,MP - 1301932,0.8

SIGN,GIVE WAY,BIA-AIA013-P,MP,MP - 1301948,-0.3

Table

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Asset Type** | **Sign Type** | **Route** | **Ref Post Type** | **Ref Post Item Label** | **Ref Post Offset** |
| **SIGN**  Asset will be loacted 0.2 from the start of the route | **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** |

**Negative Ref Post Offset**

Figure

File Column Settings

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

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **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



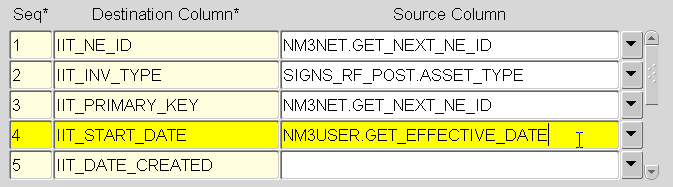
START\_DATE

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

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

|  |  |  |
| --- | --- | --- |
| **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 Abbrevaition 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 Asse 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

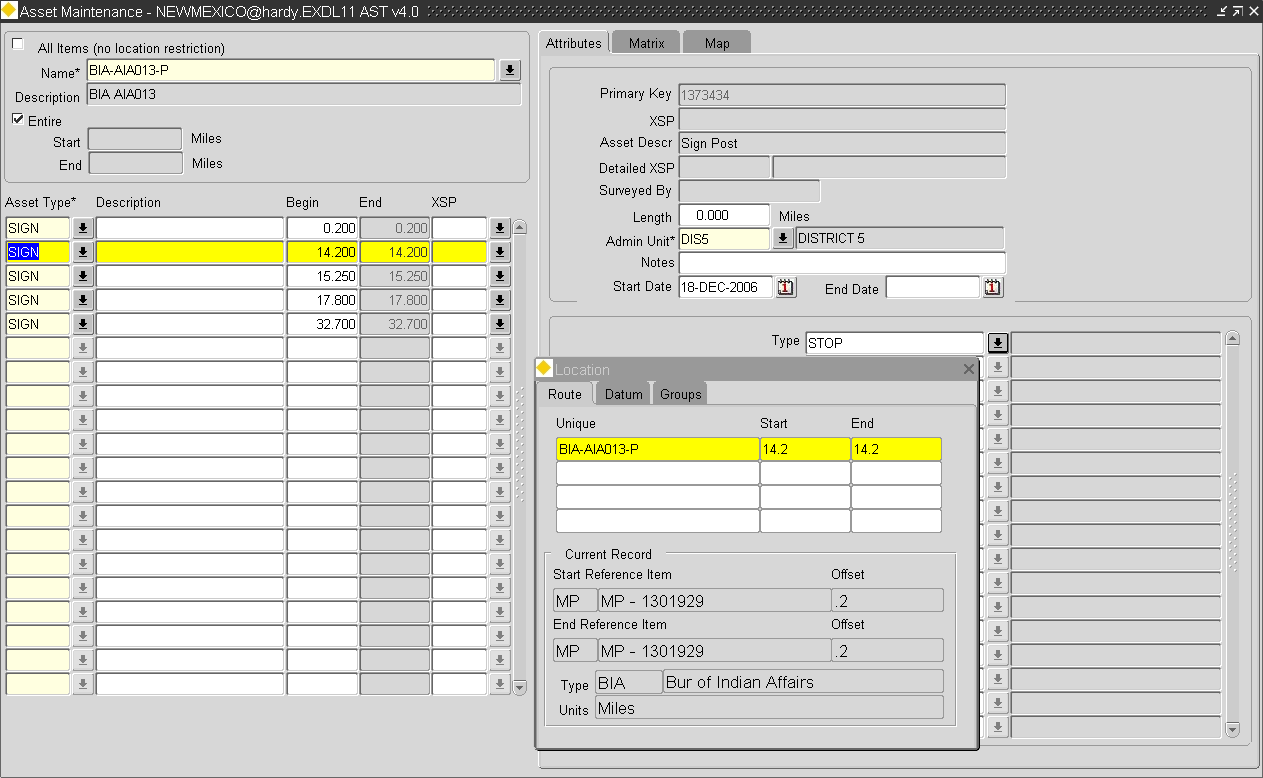
This column should be left blank when loading Point Asset Types.

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.

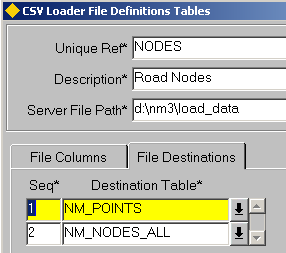
Figure 28 shows the loadd Signs within the **Asset Maintenance – NM0590** module.

Figure

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

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.

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

**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**

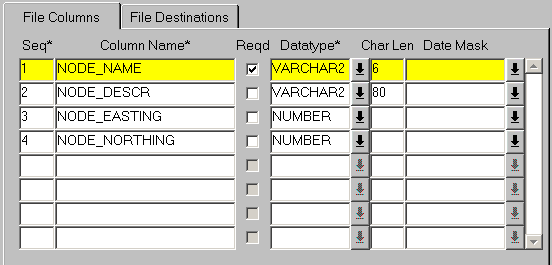
**000005,Glebe Road Int with Back Lane,583909,1679899**

**000006,Int Harrys Road and A1,581597,1681594**

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **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



File Column Settings

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

Table

|  |  |  |  |
| --- | --- | --- | --- |
| **Seq** | **Column Name** | **Datatype** | **Char Len** |
| **1** | **NODE\_NAME** | **VARCHAR2** | **6** |
| **2** | **NODE\_DESCR** | **VARCHAR2** | **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.

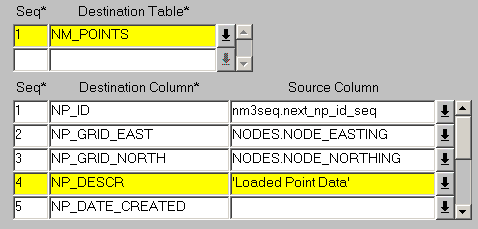
NP\_GRID\_NORTH

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



shows the complete 'mapping' of the Source Columns to the Columns in the **NM\_POINTS** Destination Table used in this example.

Table

|  |  |
| --- | --- |
| **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 Abbrevaition 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.

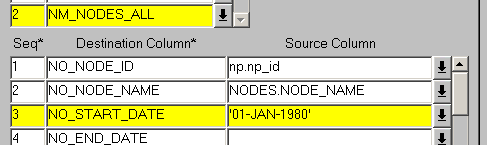
NO\_START\_DATE (Required)

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'. shows an example of this in use.

Figure



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 Abbrevaition 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 a as 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

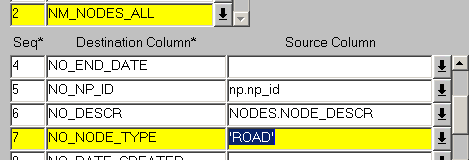


Table 27 shows the complete 'mapping' of the Source Columns to the Columns in the **NM\_NODES\_ALL** Destination Table used in this example

Table

|  |  |
| --- | --- |
| **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 46.

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 :

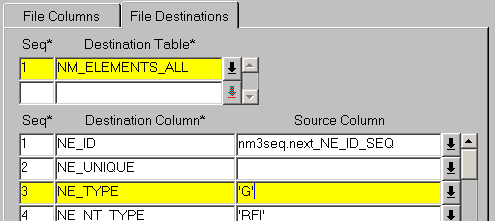
**S** Datum Elements

**G** Groups of Sections

**P** Groups of Groups

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

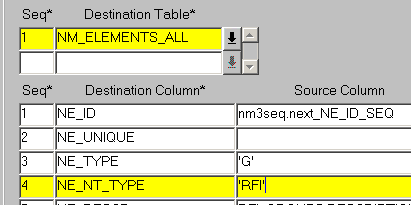


NE\_NT\_TYPE (Required)

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



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.

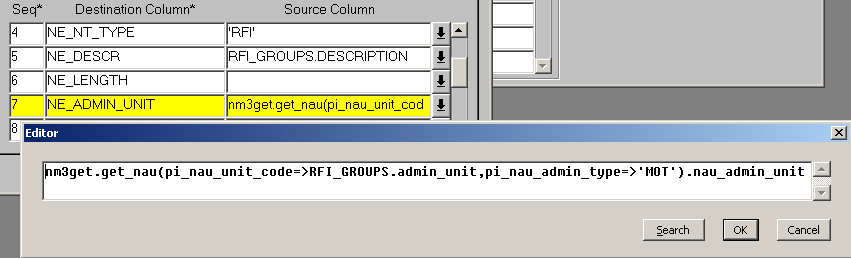
nm3get.get\_nau(pi\_nau\_unit\_code=>rfi\_groups.admin\_unit,pi\_nau\_admin\_type=>'MOT').nau\_admin\_unit

where **rfi\_groups** is the Unique Ref () 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.

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

Figure

Figure

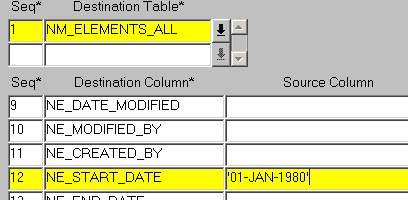
NE\_START\_DATE

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.***

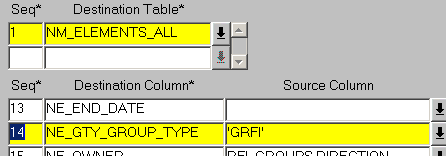
Alternatively a constant value may be added for all Element Start Dates by entering the required Date enclosed in single quotes, e.g. '01-JAN-1980'. shows an example of this in use.

Figure



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'. shows an example where Groups of Group Type GRFI are being loaded.

Figure

NE\_NO\_START and NE\_NO\_END

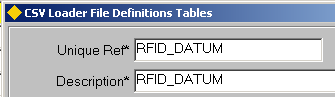
This is the internal Node id of the Start Node of the Element. Nodes are generally associated with Datum Elements but may also be associated with Groups if modelled as so in the ***Network Types - NM0002*** module (refer to the Network Manager System Admin Guide).

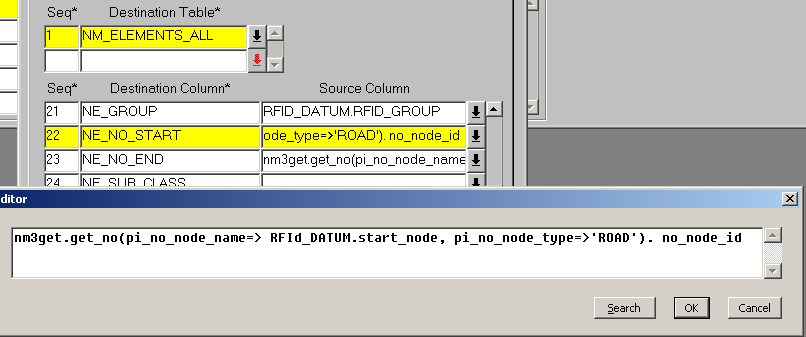
The Insert procedures used when transferring the data from the Holding Table to the Destination Table expects the internal id of the Node Point to be supplied. Since this value is unlikely to be known it is more common for the Node Name to be supplied for each record within the data file to be loaded. 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 () of the current File Definition Table, **ROAD** is the Node Type (***note that this will be the NodeType 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.

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

Figure

Figure

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'.

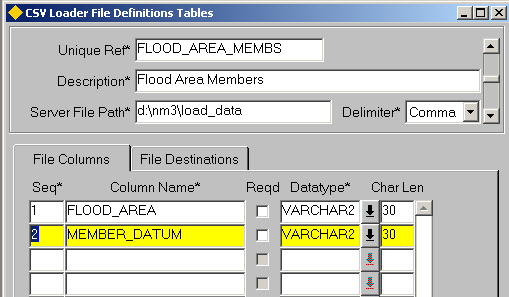
Default Attribute Values

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.

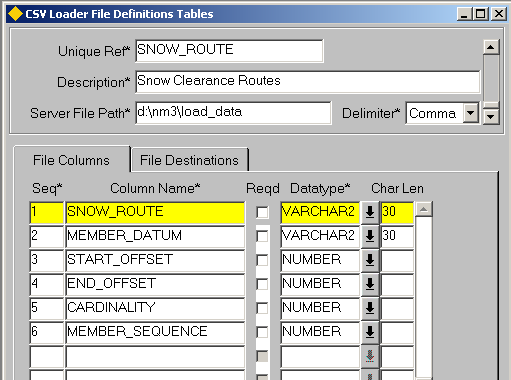
Loading Group Members

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

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 are or derived using function calls in the Source Column of the 'File Destinations' panel as described in the next section.

Figure

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 are 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.

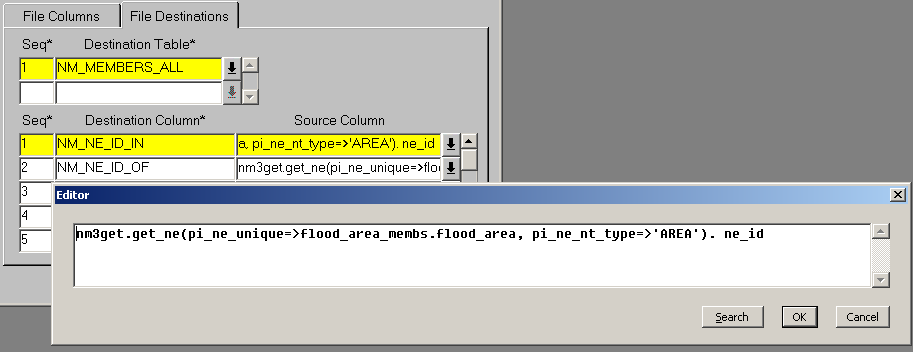
nm3get.get\_ne(pi\_ne\_unique=>flood\_area\_membs.flood\_area, pi\_ne\_nt\_type=>

'AREA'). ne\_id

where **flood\_area\_membs** is the Unique Ref () 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.

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

Figure



Figure

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

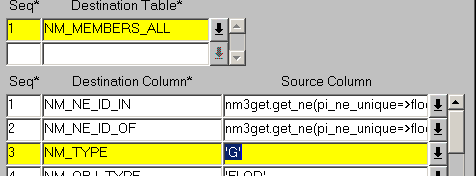
where **flood\_area\_membs** is the Unique Ref () of the current File Definition Table, **member** is the Column Name in the Holding Table which stores the value and **ROAD** is the Network Type of the member Element.

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**'

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

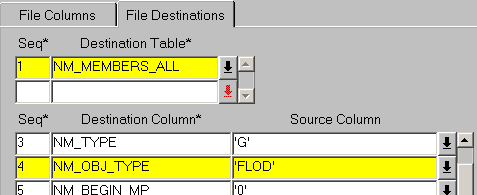
Figure



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. shows an example where member Elements are being loaded into a Groups of Type of 'FLOD'.

Figure



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.

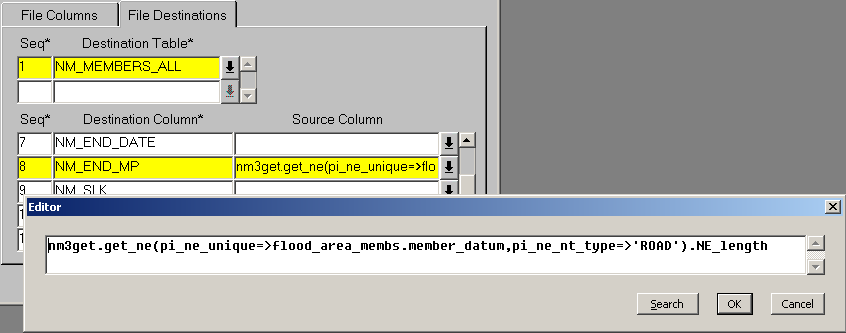
nm3get.get\_ne(pi\_ne\_unique=>flood\_area\_membs.member\_datum,

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..

shows a working example of this.

Figure



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

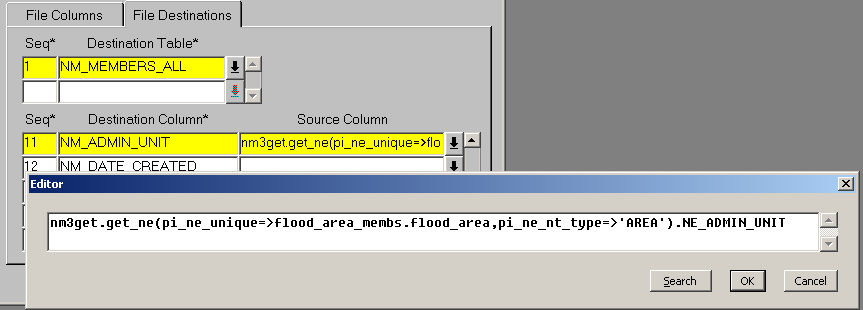
This is the Admin Unit of the Parent Group as defined in the NM\_NE\_ID\_IN Column. This would not often be included in the load file and may be derived using a function call as shown below

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..

shows a working example of this.

Figure

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

CSV Loader

Menu Option

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

* Load/Process Data - HIGWEB2030

Load/Process Data - HIGWEB2030

Figure

**Basic Data Type**

**Validation**

**CSV Data**

**File**

**.Bad**

**File**

**Holding**

**Table**

**Insert Procedure**

**and Validation**

**.Bad**

**File**

Exor Database

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 appropraite 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 :

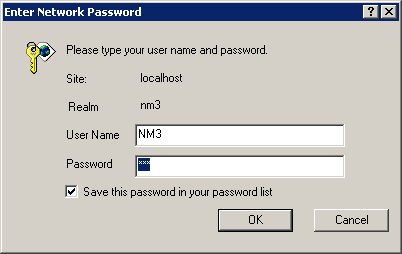
* Max and Min Values are within defined ranges
* Domain Lookup values exist
* The location at which the Item is being loaded exists, if appropriate
* The location has not already been assigned to another Admin Unit
* No Cross Attribute or Cross Item validaton Rules have been broken.

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 where detected during the initial load into the Holding Table.

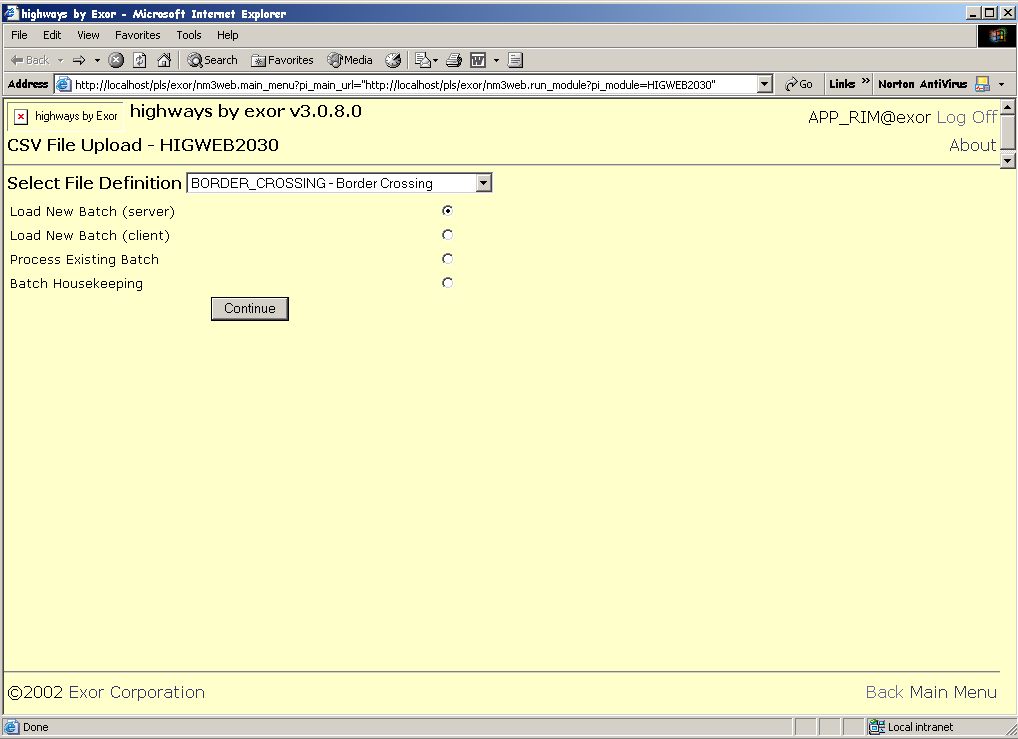
Note that the presence of erroneous data will not prevent any valid data from being successfully transferred.

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

Figure

Logging onto a Web Session

When either the ***Web Main Menu - NMWEB0000*** or the ***Load/Process Data - HIGWEB2030*** optionis 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

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)

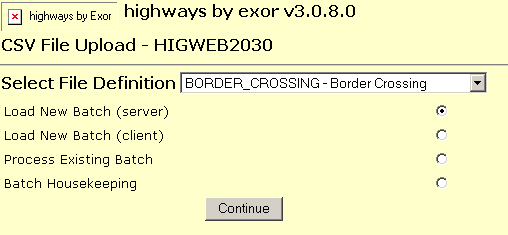
Calls <http://www.exorcorp.com/>

Back (Hyperlink)

Goes back to previous page

Main Menu (Hyperlink)

Calls the Web Main Menu - NMWEB0000

Figure

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.

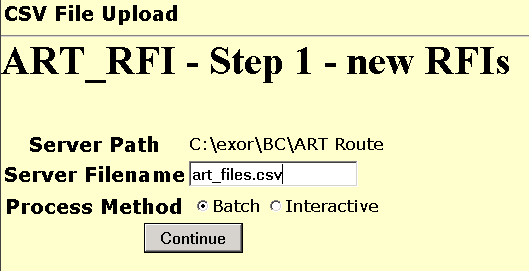
Process Existing Batch (Radio Button)

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.

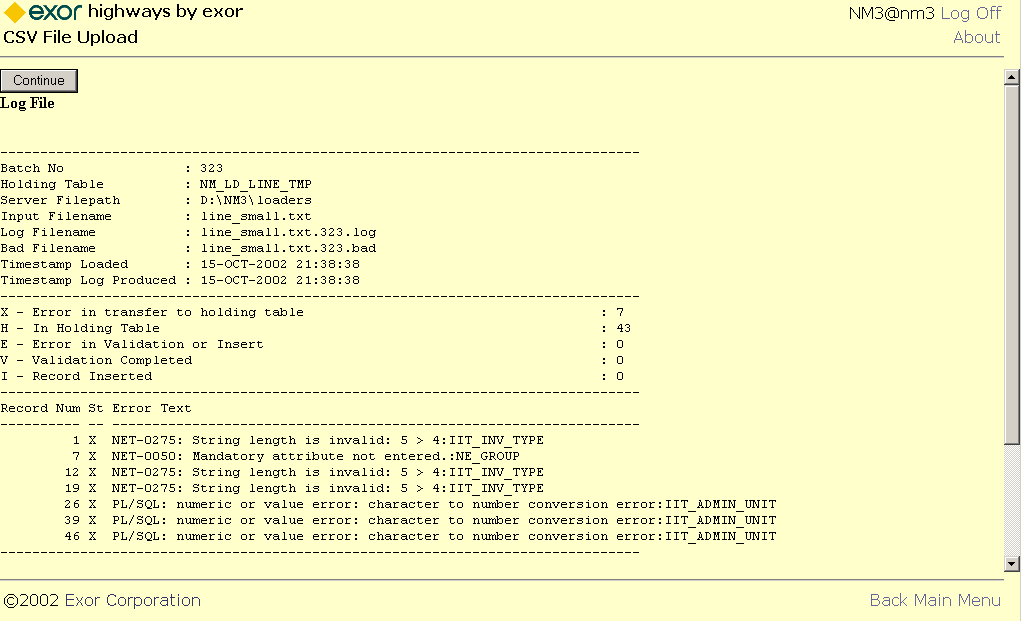
Figure

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

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

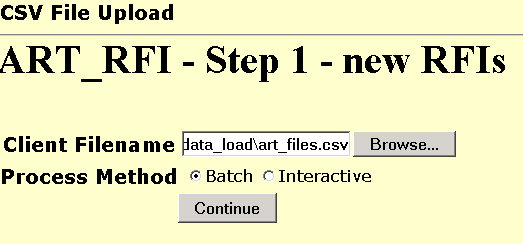
If any errors are detected during the data validation an error message will be displayed for each erroneous record. The Record Number and Status are included for each record. The Erroneous records are written to the BAD file. The table below shows some examples of typical errors which may occur.

|  |  |  |
| --- | --- | --- |
| **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 contine the current Load Batch press the **[Continue]** button at the top of the Web Page. If the original CSV data file is to be recorrected and resubmitted without finsihing the current Batch press the **[Back]** hyper link, correct the data file and press **[Continue]** to reload.

Refer to page 64 for information on the second stage of the load process.

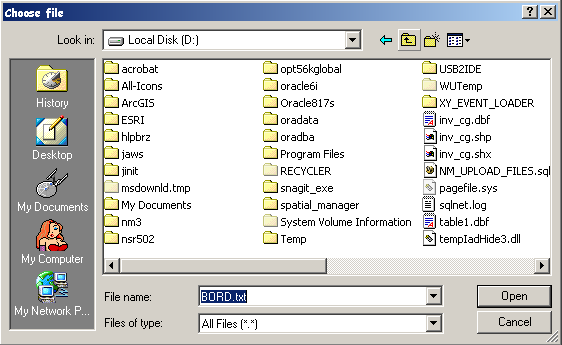
Figure

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

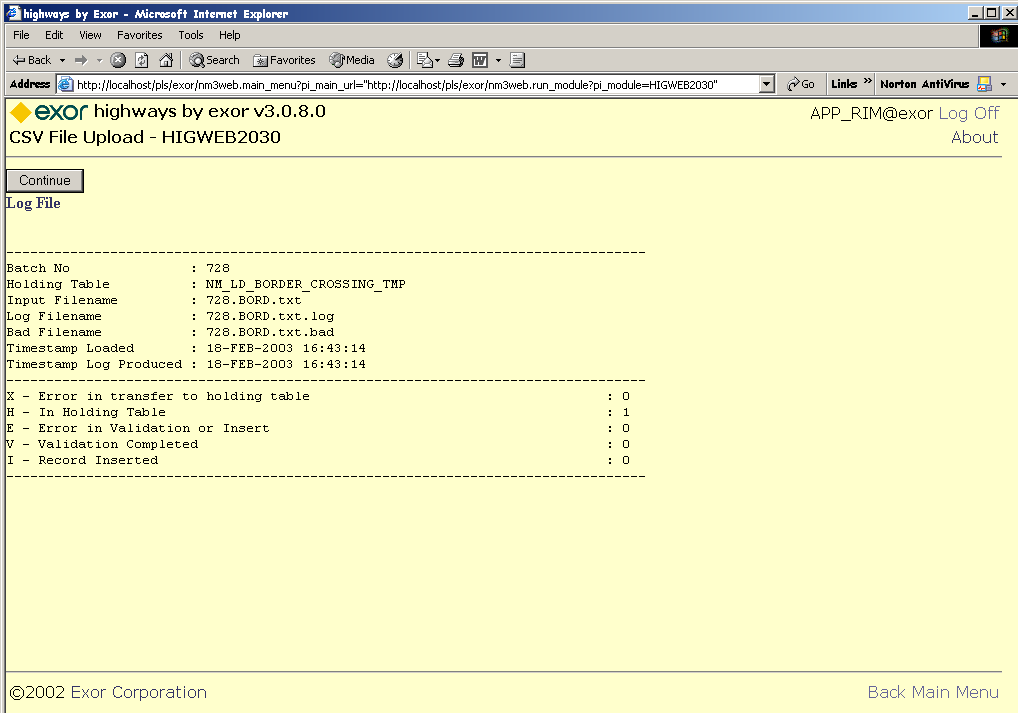
Once the full path and filename have been entered select a process method either Batch or Interactive. Selecting ‘Batch’ will initiate the load and then send an email when the load is complete. Selecting ‘Interactive’ will initiate the load and display results when the load is complete. If you select ‘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 databse 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.

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

Figure

The LOG File contains the following information:

* Batch No
* Holding Table Name
* Input Filename (with apended 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

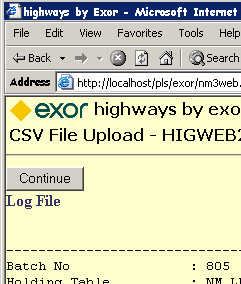
If any errors are detected during the data validation an error message will be displayed for each erroneous record. The Record Number and Status are included for each record. The Erroneous records are written to the BAD file. The table below shows some examples of typical errors which may occur.

|  |  |  |
| --- | --- | --- |
| **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 (). This is a hyper Link to the file.

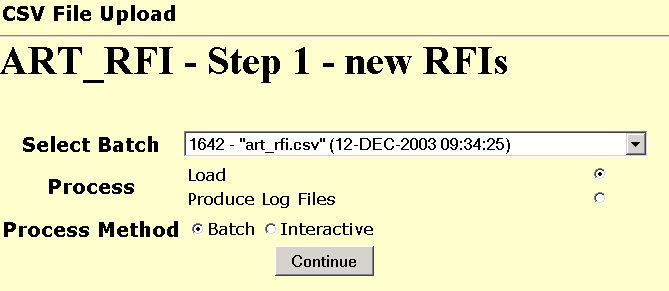
Figure



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

To contine 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 64 for information on the second stage of the load process.

Figure

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.

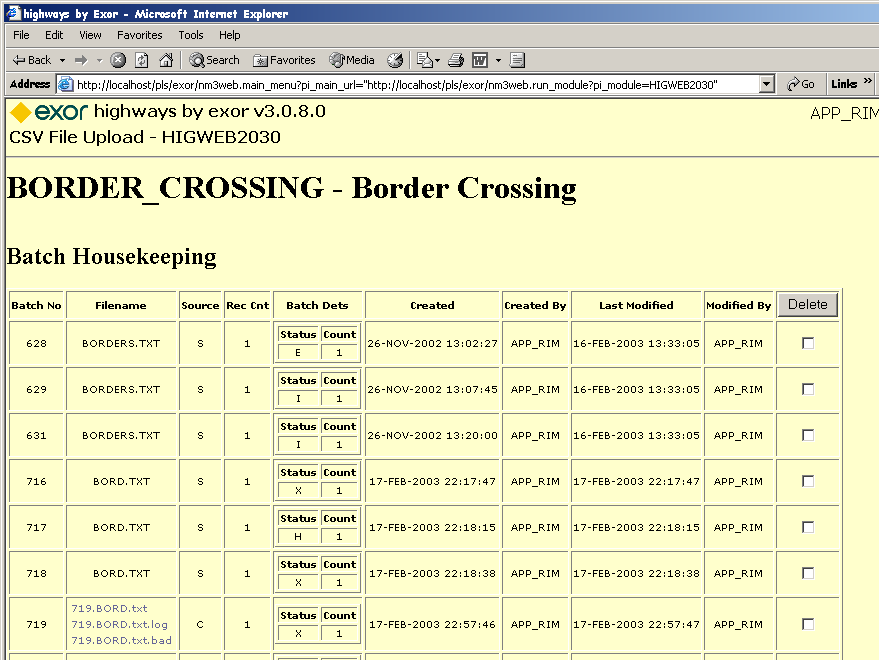
Produce Log Files (Radio Button)

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



Batch Housekeeping

Previously loaded batched may be removed from the system using the Batch Houskeeping 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 · 66

C

CSV File Upload Process · 55

***CSV Loader Destination Table*** · ***6***, 8, 18, 21

CSV Loader File Definitions Table · 5, 9, ***10***, ***11***, ***12***, ***13***, 14, 15, 16, 17, 58

***CSV Loader Files Definitions Tables - HIG2020*** · ***6***, 7, 8, 55, 56, 60

D

Default Values Panel · 8

Destination Table · 6, 7, 8, 9, ***10***, ***11***, ***12***, ***13***, 14, 15, 16, 17, 21, 22, 23, 25, 26, 27, 28, 33, 34, 35, 39, 40, 41, 42, 44, 45, 47, 48, 56, 57, 59, 65

Destination Tables · 5, 6, 7, 9, ***11***, ***12***, 14, 15, 16, 18, 21, 23, 25, 31, 33, 37, 39, 56, 65

F

File Column Settings · 24, 32, 38, 48

File Columns · 19, 20, 21, 24, 32, 38, 42

File Columns Panel · 19

File Definitions Table · 18

File Destination Settings · 25, 33, 39

File Destinations · 6, 8, 21, 42, 47, 48, 56

File Destinations Panel · 21

H

***HIGWEB2030*** · ***54***, ***55***, ***57***, 66

Holding Table · 5, 7, 17, 18, 19, 20, 21, 24, 25, 26, 27, 28, 29, 32, 33, 34, 35, 36, 38, 39, 40, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 55, 56, 57, 58, 60, 61, 63, 65

I

Insert Procedure · 7, 9, ***10***, ***11***, ***12***, 14, 15, 16

L

Load / Validate Data (2nd Stage) · 65

Load New Batch from Client · 62

Load New Batch from Server · 60

Load/Process Data · 54, ***55***, ***57***

Loading Asset · 9, 23

Loading Asset Item Data · 9

Loading Asset Item Location Data · 9

Loading Group Members · 47

Loading Network Elements · 16, 41

Loading Network Node Points · 15

Loading Network Nodes · 21, 37

Logging onto a Web Session · 57

N

NM\_NODES\_ALL · 15, 37, 39, 40, 41

NM\_POINTS · 15, 37, 39, 40

***NMWEB0000*** · ***57***, ***58***

R

Reference Post · 13, 30, 31, 35, 36

Reference Post Offset · 13, 30

Route Mile Post’ referencing · 13, 30

S

Selecting the File Definition · 58

Server File Path · 18, 56

V

***V\_LOAD\_INV\_MEM\_ELE\_MP\_AMBIG*** · ***9***, ***12***, ***13***

***V\_LOAD\_INV\_MEM\_ELE\_MP\_EXCL*** · ***9***, ***10***, ***11***

V\_LOAD\_INV\_MEM\_ON\_ELEMENT · 9, ***10***, 14, 25, 27, 33, 35

V\_LOAD\_LOCATE\_INV\_BY\_REF · 2, ***13***, 14, 35

Validation Procedure · 7, 9, ***10***, ***11***, ***12***, 56