New Zealand Info Systems Limited

NODEM: an off-the-shelf software, devloped to provide software solution for NAMS ODM Guidelines. It helps Asset Managers more informed & assist in making optimised decisions with confidence.

DRAFT

NODEM Developer's Manual

March 2006

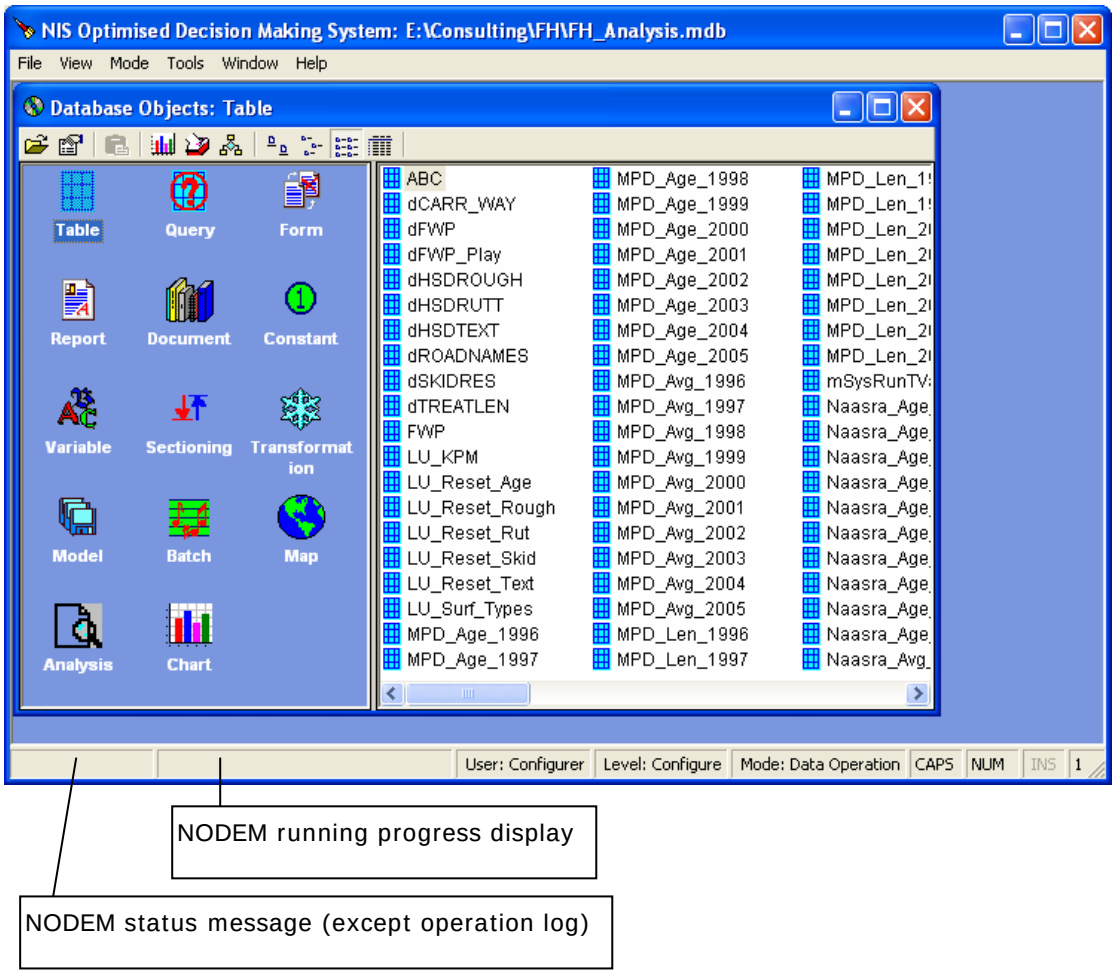
NEW ZEALAND INFO SYSTEMS LIMITED

CONTENTS

1 A GLIMPSE OF NODEM SYSTEM

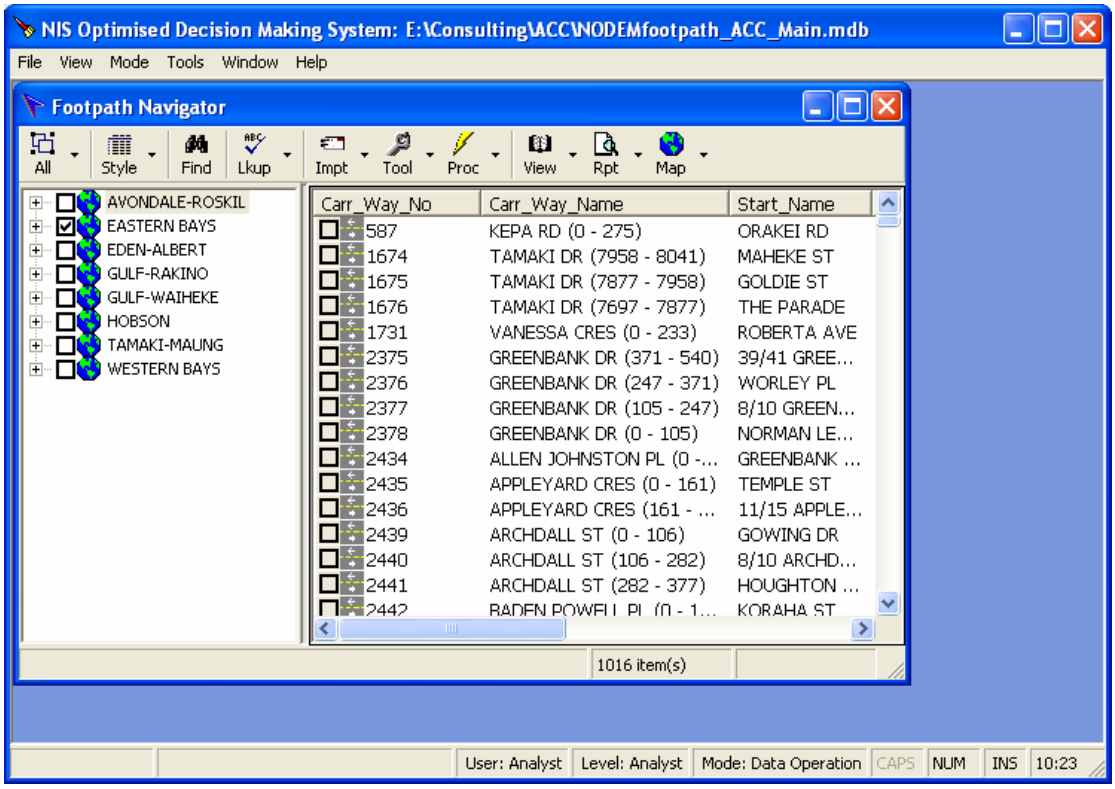
1.1 Interface for Developer (Object Browser)

Many types of data and operational objects can be created and maintained by the business system configurer to satisfy the business requirements from data capture, data integration, data processing, modelling to information reporting and thematic map presentation. A lot of operations and functions are provided to operate on NODEM objects to make a system without a single line of coding.



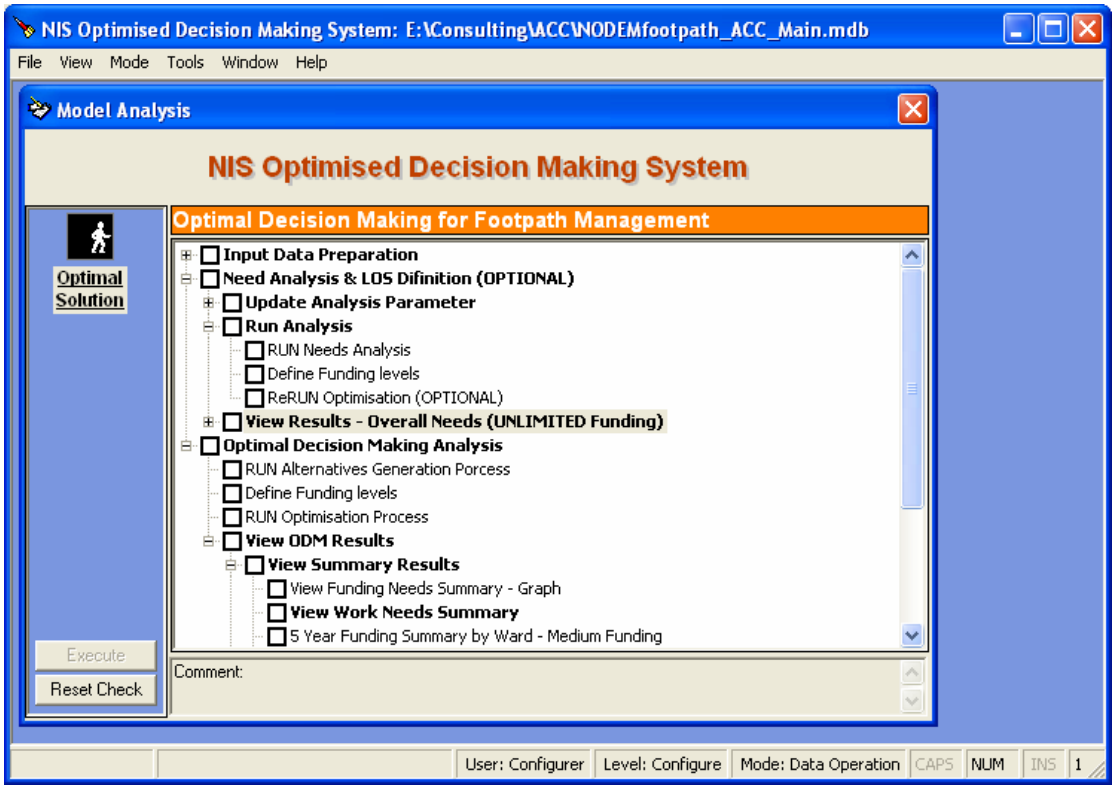
1.2 Interface for User (Navigation)

Navigation of data (can be list view or map view) and integration of functions can be customised to satisfy the business and user requirements for different business contexts and systems. NODEM can support as many as this kind of systems (to be exactly, sub-systems) based on one single database or different databases. Data operation can be filtered based on the user selection if required.



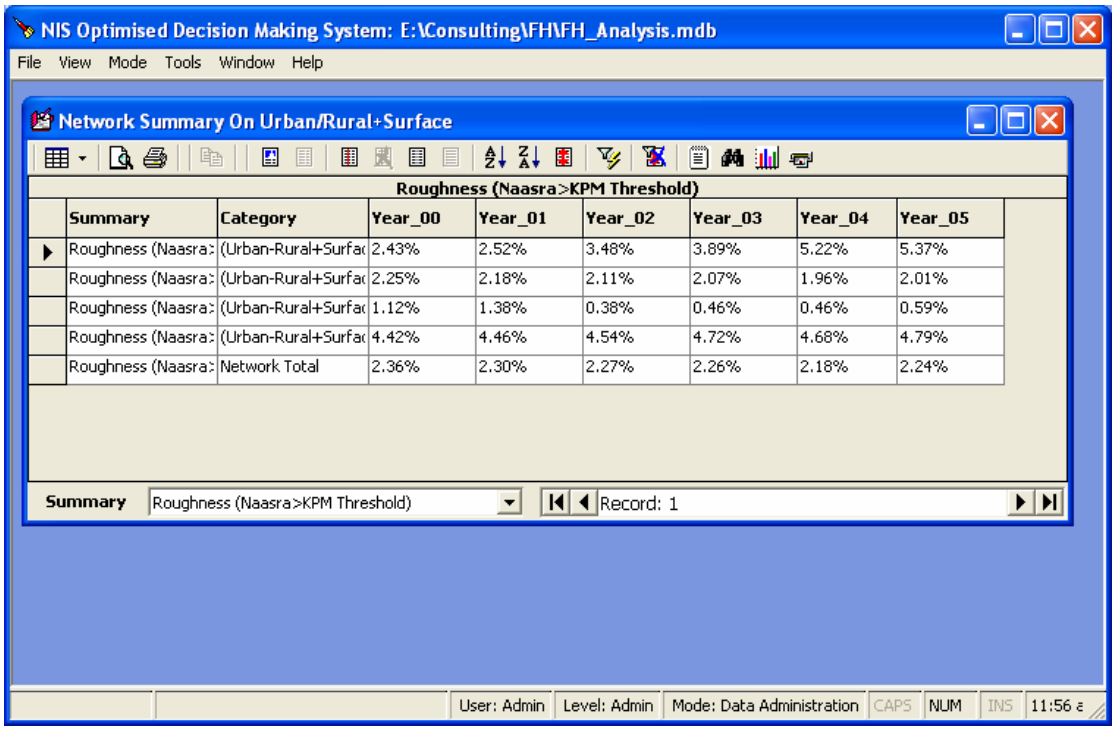
1.3 Interface for Analysis (Optional Interface)

Complicated modelling and analysis can be customised to the user's convenience and usability of the systems. Operations can be grouped in a logic manner, can be run in batch mode if no much user intervention needed or in step by step base, and can be tracked when operation is finished.



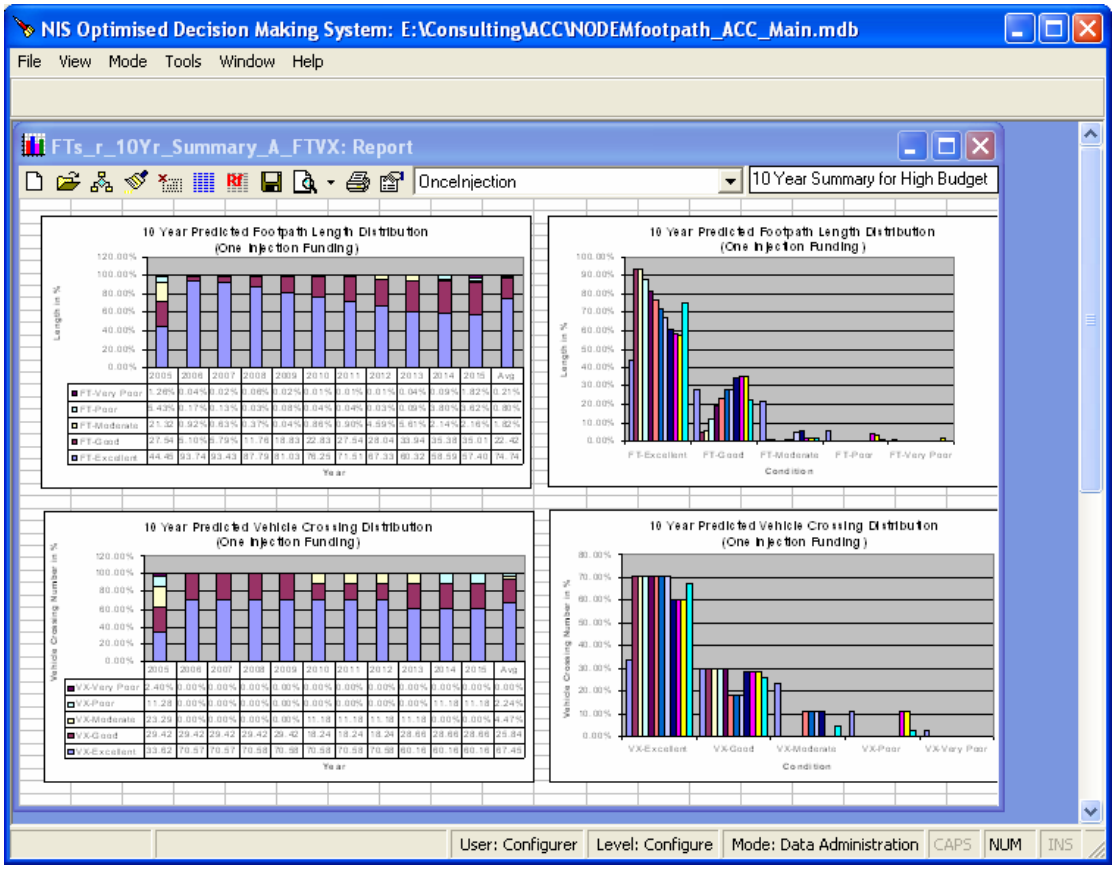
1.4 Data Manipulation and View

Data can be viewed and manipulated in default data grid format with many built-in functions in no effort.



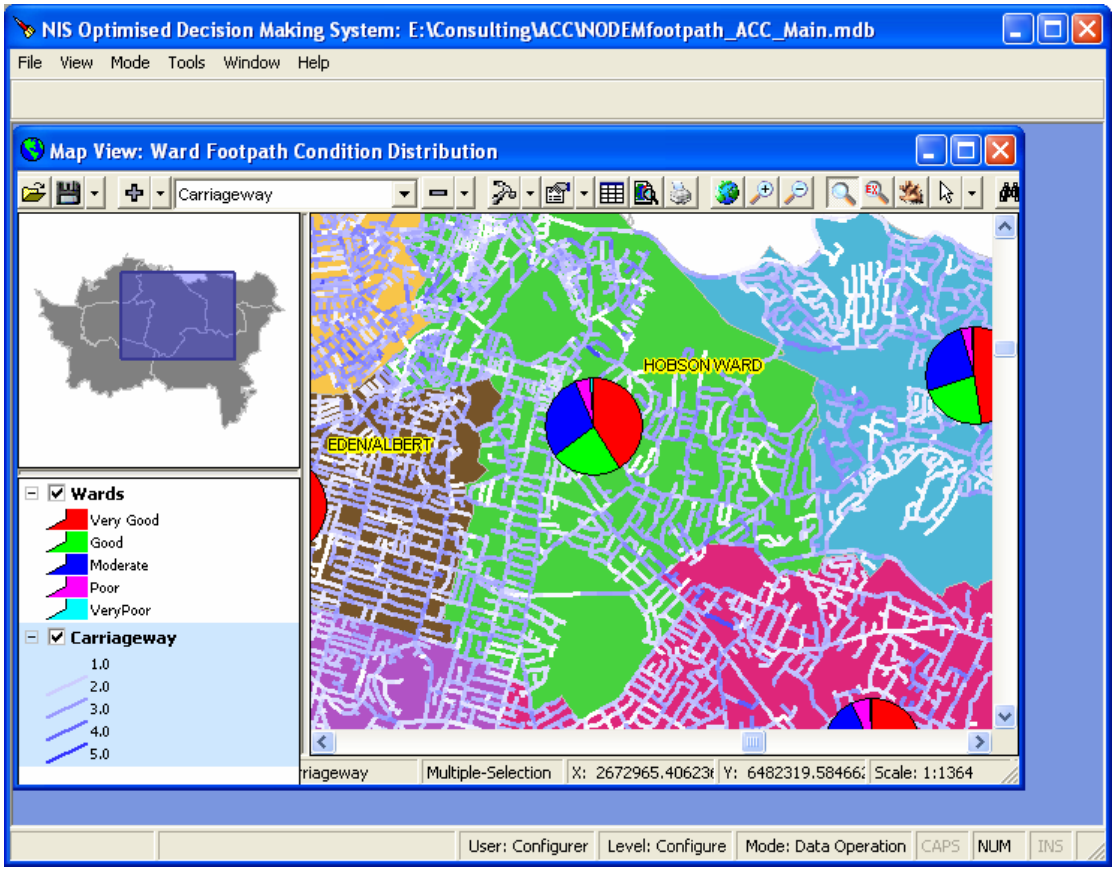
1.5 Definable Data Reports

Reports in NODEM can be seamlessly integrated with Excel with data from the database. So the user can use his Excel skill to design reports and graphs to his very requirements. Designed reports can be linked external files or embedded objects within the database for portability.



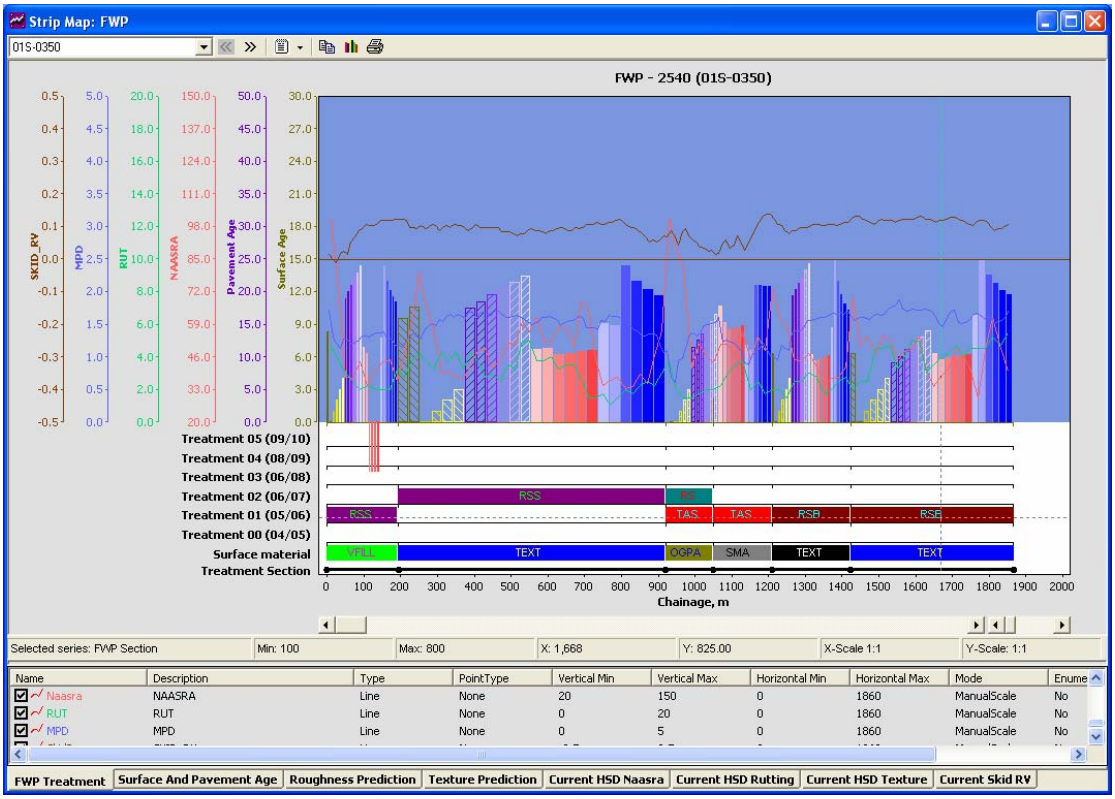
1.6 Designable Thematic Map

Map from external GIS system can be saved into NODEM database and link to internal data to be presented on NODEM map viewer. Thematic maps can thus be easily designed to represent the user interest. Designed thematic map can be viewed, printed or exported to feed other GIS software.



1.7 Definable Stored Chart for Data View

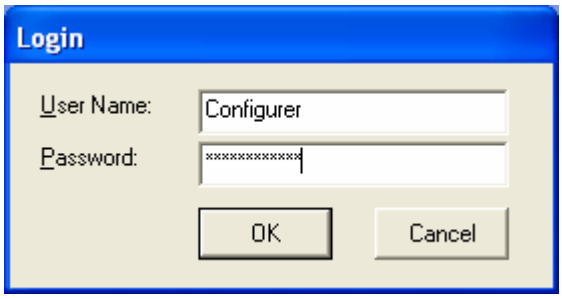
Data can be viewed (and played) through user definable charts or strip maps in a variety of mixed styles, to give the user maximised visual effects of the data condition and other information to his interest.


Strip Map: FWP
01S-0350
FWP - 2540 (01S-0350)
SKID_RV
MPD
RUT
NAASRA
Pavement Age
Surface Age
Treatment 05 (09/10)
Treatment 04 (08/09)
Treatment 03 (06/08)
Treatment 02 (06/07)
Treatment 01 (05/06)
Treatment 00 (04/05)
Surface material
Treatment Section
RSS
TAS
RSB
RSE
VTLL
TEXT
OGPA
SMA
Chainage, m
Selected series: FWP Section
Min: 100
Max: 800
X: 1,668
Y: 825.00
X-Scale 1:1
Y-Scale: 1:1
Name
Description
Type
PointType
Vertical Min
Vertical Max
Horizontal Min
Horizontal Max
Mode
Naasra
NAASRA
Line
None
20
150
0
1860
ManualScale
No
RUT
RUT
Line
None
0
20
0
1860
ManualScale
No
MPD
MPD
Line
None
0
5
0
1860
ManualScale
No
FWP Treatment
Surface And Pavement Age
Roughness Prediction
Texture Prediction
Current HSD Naasra
Current HSD Rutting
Current HSD Texture
Current Skid RV

2 USER MANAGEMENT

User account for NODEM applies to NODEM software itself. In other words, it applies the same access level to all NODEM databases.

2.1 User Login



Once user logged in, user information will be displayed in status bar.

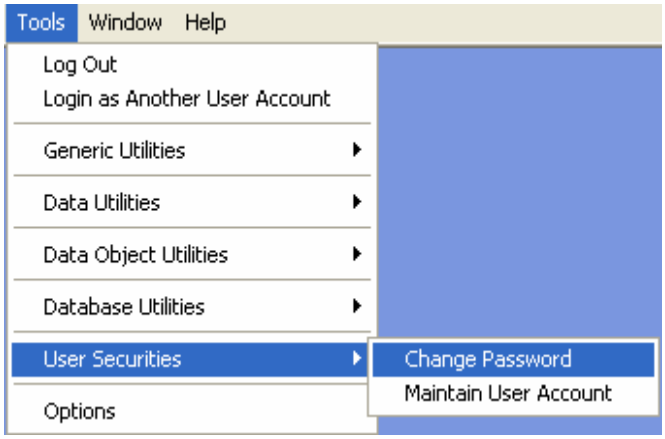
2.2 User Logout

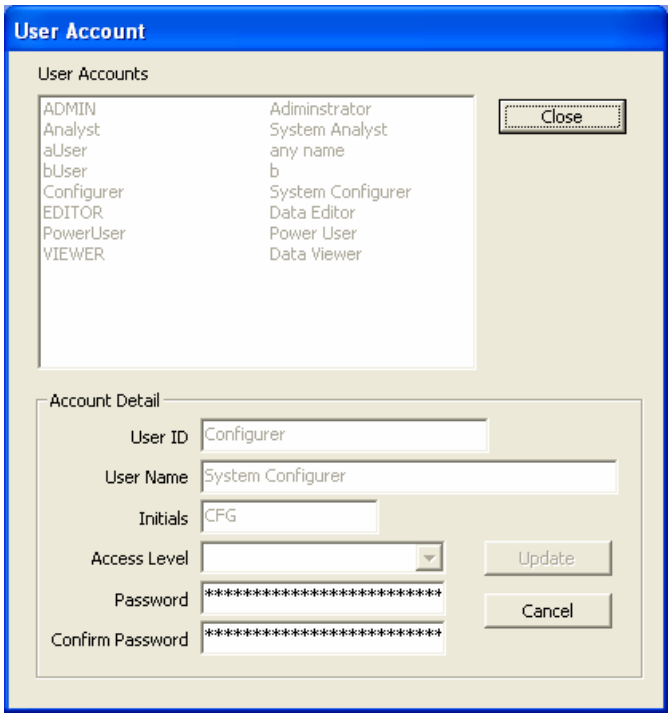
Under Tools menu, select Log Out to log out. Log out will close the database as well

Logged in user can also log in as another user

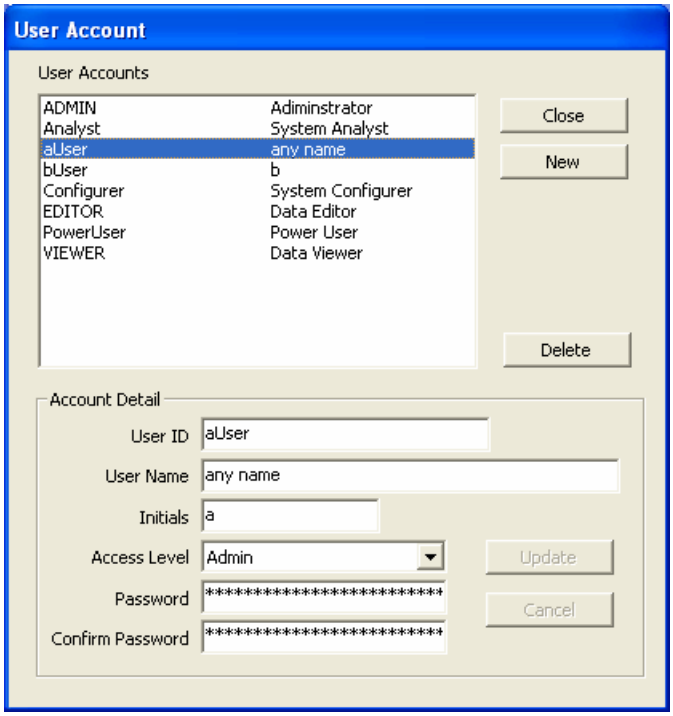
2.3 User Securities

Logged in user can change his password no matter what access level he has.





User of Admin level or above can maintain user accounts added to the system, but not to the built-in user accounts.



2.4 User Access Level

User Level	Description
Viewer	Can only view data objects
Editor	Can modify data in data object from configured (controlled) operation
Power User	Can modify data in data object and run operations
Analyst	Can modify definition in operation object
Admin	Can modify data object, data in operation object and user account
Configure	Can create, delete and modify all data objects and user account

User access to object and operations also depends on system data access mode. For detail about user access level and types of objects, please refer to section NODEM objects

3 NODEM OBJECT OPERATIONS

3.1 NODEM Objects

As an open platform, NODEM have many types of objects for system configurer to define to satisfy the business requirements. Following shows all the objects NODEM supports at the time of this manual is developed. Depending on purchase, the user may not have all objects available in his system.

Broadly all NODEM objects can be grouped into following types:

Data Object
Including table, query, form and report

Operational Object
Including sectioning, transformation, constant, variable, model, batch, analysis, chart. All operational objects are executable except constant and variable which are used inside model

Viewable Object
Including map and all data objects

Designable Object
Including table, query, form, chart, report and all operational objects

Executable Object
All operational objects are executable except constant and variable which are used inside model

Document Object
Text document for user logging and documentation.

Other Object
This refers to all objects attached to some of the above objects, including database trigger (which can only attach to table, query and form), runtime (directive) variable (which can be used in operational objects for runtime input and calculation), infowz (information wizard, which can only attach to model and batch for collecting preset runtime variables' value)

3.2 NODEM Object Operations

The operation available depends on type of object, system operation mode, level of user access. Broadly NODME supports a list of operations as following:

New: create new object

Open: open an existing object

Design: design an existing object

Property: property of existing object (not supported yet)

Copy: copy an existing object

Cut: cut an existing object

Paste: paste copied or cut object

Graph: simple data graph on data object (table and data query only)

Validate: validate data object (table and data query only)

Relationship: relationship of objects (not supported yet)

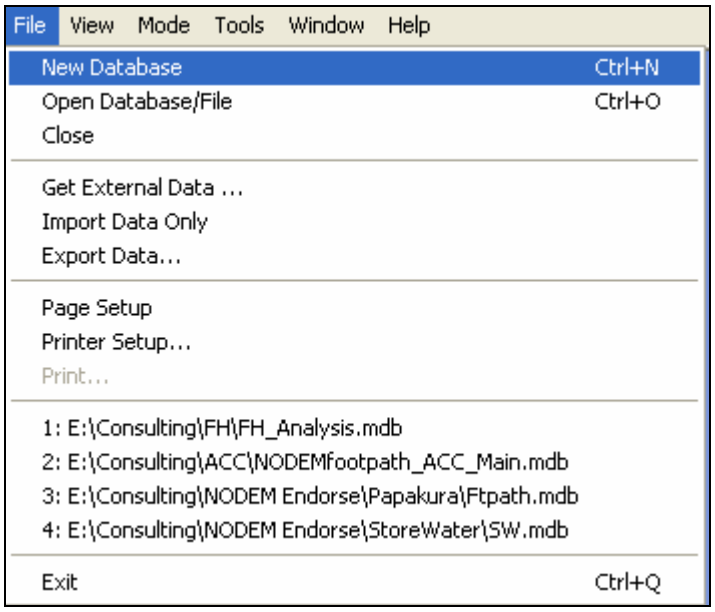
Trigger: define database trigger on data change (table, data query and form only. Form not supported yet)

InfoWizard: start wizard for operational object runtime (directive) variable preset value

Run: run executable objects

3.3 NODEM Menu Level Functions

A list of functions in NODEM environment system menu level are supplied with NODEM. But their availability is again dependent on user level and access mode. Following are a list of functions:



File Menu

Create new database

Open database or file

Close database or file

Import/link to external data

Import external data

Export internal data

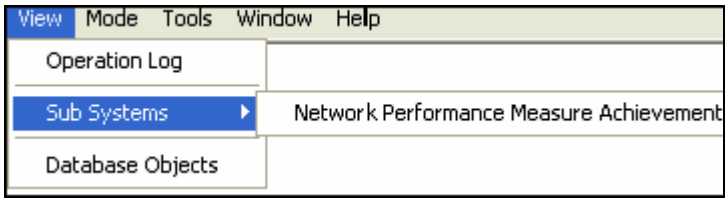
Page setup

Printer setup

Most recent opened data or file

Exit NODEM

View Menu

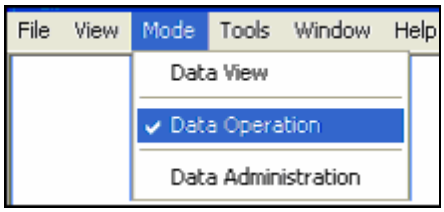


Open operation log

Open configured system

Open object browser

Mode Menu

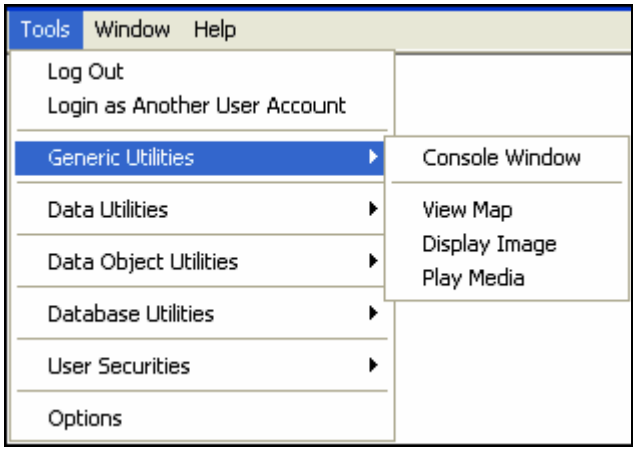


Change to View mode

Change to Operation Mode

Change to Admin mode

Tools Menu



Log out NODEM

Log in as another user

Generic utilities

Data utilities for data proc

Data object utilities

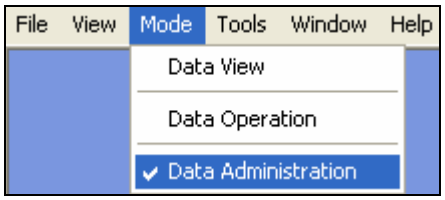
Database utilities

User securities and account

Genera options

3.4 NODEM Operational Mode

There are 3 operational modes in NODEM. Depending on the user access level, some mode(s) may not be available for the user to switch over. The 3 modes are:



Data Administration
For user with access level above Analyst (ie Admin or Configure) to administer the system, data objects and operational objects, and other system functions

Data Operation
For user with access level above Editor to run and maintain data and operational objects to some extent. For an already configured system, it will support nearly most of the exposed operations.

Data View
For user with access level of Viewer and Editor. Data may be editable in controlled context (ie the operation is configured so, such as Edit function inside Navigator) for Editor even in data view mode.

A user of higher level access can limit his own operation on the system by switch to lower level of operational mode available.

3.5 NODEM Object Operation Matrix

Operations depend on user access level and operational mode. Following matrix describes the relationships among user level, operational mode and type of objects.

	Viewer (or Data View Mode)						
	Create	**Open (View)**	**Open (Edit)**	**Design**	**Delete**	**Run**	**Validation**
Table		X					
Query		X					
Form		X					
Report		X					
Document	X	X	X		X		
Constant							
Variable							
Sectioning							
Transformation							
Model							
Batch							
Map		X					
Analysis							
Chart						X	
Get External Data							
Import Data Only							
Data Utilities							

Data Object Utilities							
Database Utilities							
Maintain User Account							

Editor (or Data View Mode)							
	Create	**Open (View)**	**Open (Edit)**	**Design**	**Delete**	**Run**	**Validation**
Table		X	X*				
Query		X	X*				
Form		X	X*				
Report		X					
Document	X	X	X		X		
Constant							
Variable							
Sectioning							
Transformation							
Model							
Batch							
Map		X					
Analysis							
Chart						X	
Get External Data							
Import Data Only							
Data Utilities							
Data Object Utilities							
Database Utilities							
Maintain User Account							

** Data can be edited only in configured context, not from data object browser.*

Power User (or Data Operation Mode)							
	Create	**Open (View)**	**Open (Edit/Run)**	**Design**	**Delete**	**Run**	**Validation**
Table		X	X				X
Query		X	X				X
Form		X	X				
Report		X	X	X			
Document	X	X	X		X		
Constant							
Variable							
Sectioning						X	
Transformation						X	
Model						X	
Batch						X	
Map		X	X	X			
Analysis						X	
Chart						X	
Get External Data							
Import Data Only						X	
Data Utilities						X	
Data Object Utilities							
Database Utilities						X	
Maintain User Account							

Analyst (or Data Operation Mode)

	Create	Open (View)	Open (Edit/Run)	Design	Delete	Run	Validation
Table		X	X				X
Query		X	X				X
Form		X	X				
Report	X	X	X	X	X		
Document	X	X	X		X		
Constant		X	X	X			
Variable		X	X	X			
Sectioning		X	X	X		X	
Transformation		X	X	X		X	
Model		X	X	X		X	
Batch		X	X	X		X	
Map	X	X	X	X	X		
Analysis		X	X	X		X	
Chart		X	X			X	
Get External Data						X	
Import Data Only						X	
Data Utilities						X	
Data Object Utilities						X	
Database Utilities						X	
Maintain User Account							

Admin (on Data Operation Mode)							
	Create	Open (View)	Open (Edit/Run)	Design	Delete	Run	Validation
Table		X	X				X
Query		X	X				X
Form		X	X				
Report	X	X	X	X	X		
Document	X	X	X		X		
Constant		X	X	X			
Variable		X	X	X			
Sectioning		X	X	X		X	
Transformation		X	X	X		X	
Model		X	X	X		X	
Batch		X	X	X		X	
Map	X	X	X	X	X		
Analysis		X	X	X		X	
Chart		X	X			X	
Get External Data						X	
Import Data Only						X	
Data Utilities						X	
Data Object Utilities						X	
Database Utilities						X	
Maintain User Account						X	

Admin (on Data Administration Mode)							
	Create	Open (View)	Open (Edit/Run)	Design	Delete	Run	Validation
Table	X	X	X	X	X		X
Query	X	X	X	X	X		X

Form	X	X	X	X	X		
Report	X	X	X	X	X		
Document	X	X	X		X		
Constant		X	X	X			
Variable		X	X	X			
Sectioning		X	X	X		X	
Transformation		X	X	X		X	
Model		X	X	X		X	
Batch		X	X	X		X	
Map	X	X	X	X	X	X	
Analysis		X	X	X		X	
Chart		X	X	X		X	
Get External Data						X	
Import Data Only						X	
Data Utilities						X	
Data Object Utilities						X	
Database Utilities						X	
Maintain User Account						X	

**Additional function available for Admin to load external GIS data into database.*

Configure (on Data Administration Mode)							
	Create	**Open (View)**	**Open (Edit/Run)**	**Design**	**Delete**	**Run**	**Validation**
Table	X	X	X	X	X		X
Query	X	X	X	X	X		X
Form	X	X	X	X	X		
Report	X	X	X	X	X		
Document	X	X	X		X		
Constant	X	X	X	X	X		
Variable	X	X	X	X	X		
Sectioning	X	X	X	X	X	X	
Transformation	X	X	X	X	X	X	
Model	X	X	X	X	X	X	
Batch	X	X	X	X	X	X	
Map	X	X	X	X	X	X	
Analysis	X	X	X	X	X	X	
Chart	X	X	X	X	X	X	
Get External Data						X	
Import Data Only						X	
Data Utilities						X	
Data Object Utilities						X	
Database Utilities						X	
Maintain User Account						X	

**Additional functions available for configurer to design Trigger, run Infowz, and define subsystems (under Tools|Database Utilities|Define Sub Systems)*

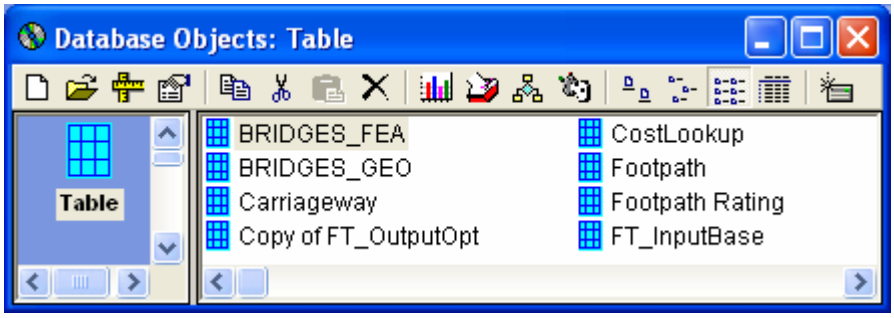
Please note, sometimes operations may be run implicitly at a lower level. For instance, a trigger may be run with a user of Editor level; a table may get deleted at a user of Power User level from a batch. This is to ensure the configured operation can be run smoothly with a careful design of the system configurer.

4 WORKING WITH TABLES

Data tables are used to keep static data in the database.

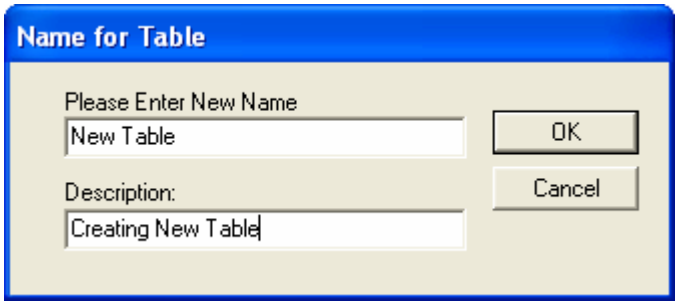
Database Objects: Table functionality available in NODEM to create, modify and delete the existing tables.

A list of all the Tables can be viewed in **Database Objects: Table** window.



4.1 Create Table

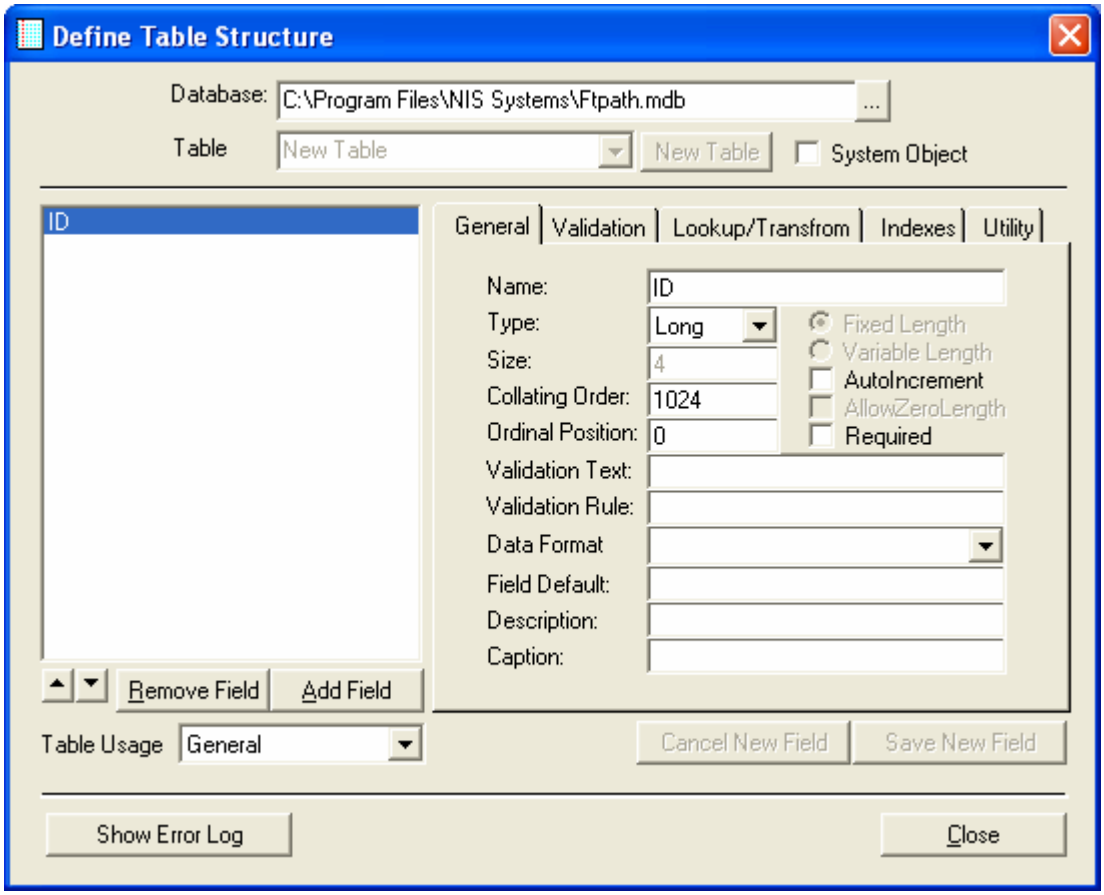
A new **Table** can be created by clicking on  button in the top menu in the **Data Objects: Table** window. **Name for Table** window allows defining the name and description of the Table.



4.2 Design Table Structure

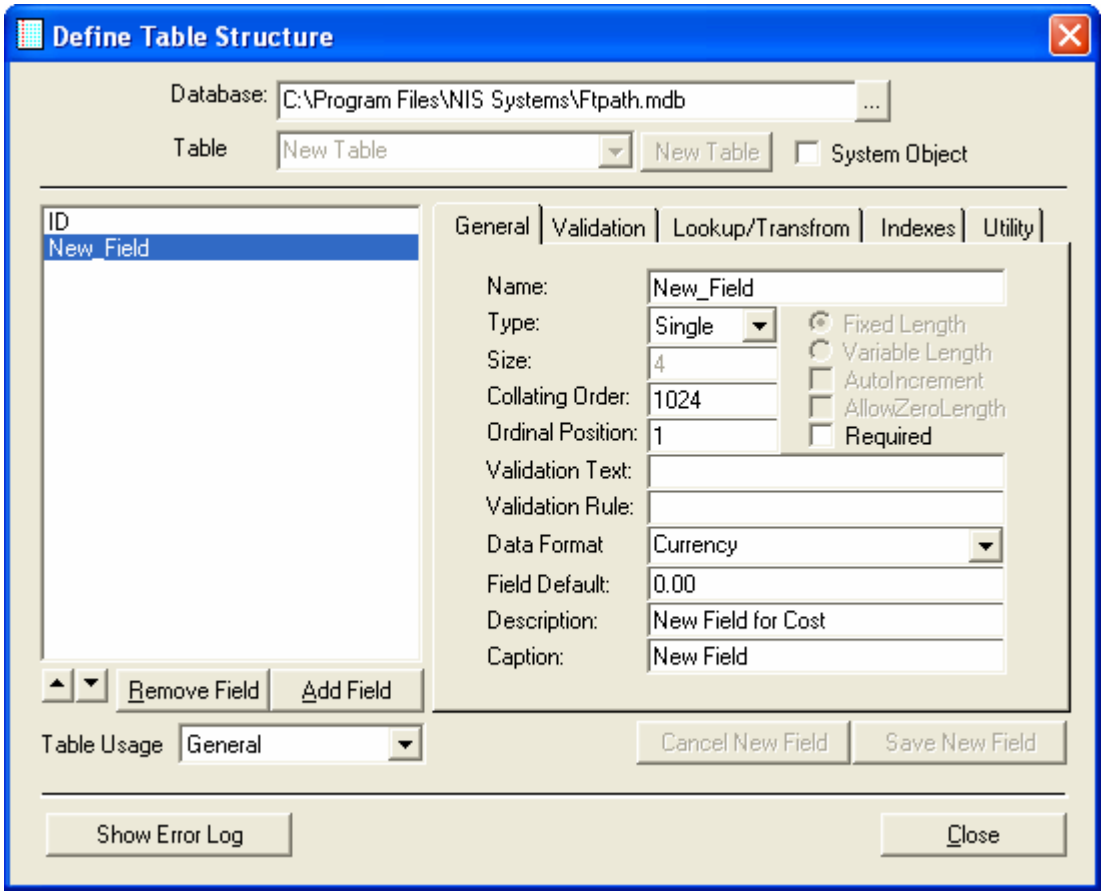
A new data field can be added just clicking on 

button.



The minimum definition required to create a data field include:

Name: Data Field Name
Type: Data Field Type
Size: Size of the Text Field
Description: A Description of the field. This will be displayed in the table header when a table is opened. The examples below show the table without and with the descriptions defined
Caption: This defaults to the field Description above. When a data input form is generated for the table



4.3 Define Extended Field Properties

Other specific fields requiring definition include:

Required: If checked, the field is required entry
Allow Zero Length: If checked, Text field can have zero length
Validation Text: The text to display if the field fails validation
Validation Rule: The rule used to validate the data when input or imported
Data Format: Allow to select a data format option
Field Default: The default value to apply to the field during data entry or importing

4.4 Define Table Validation Rule

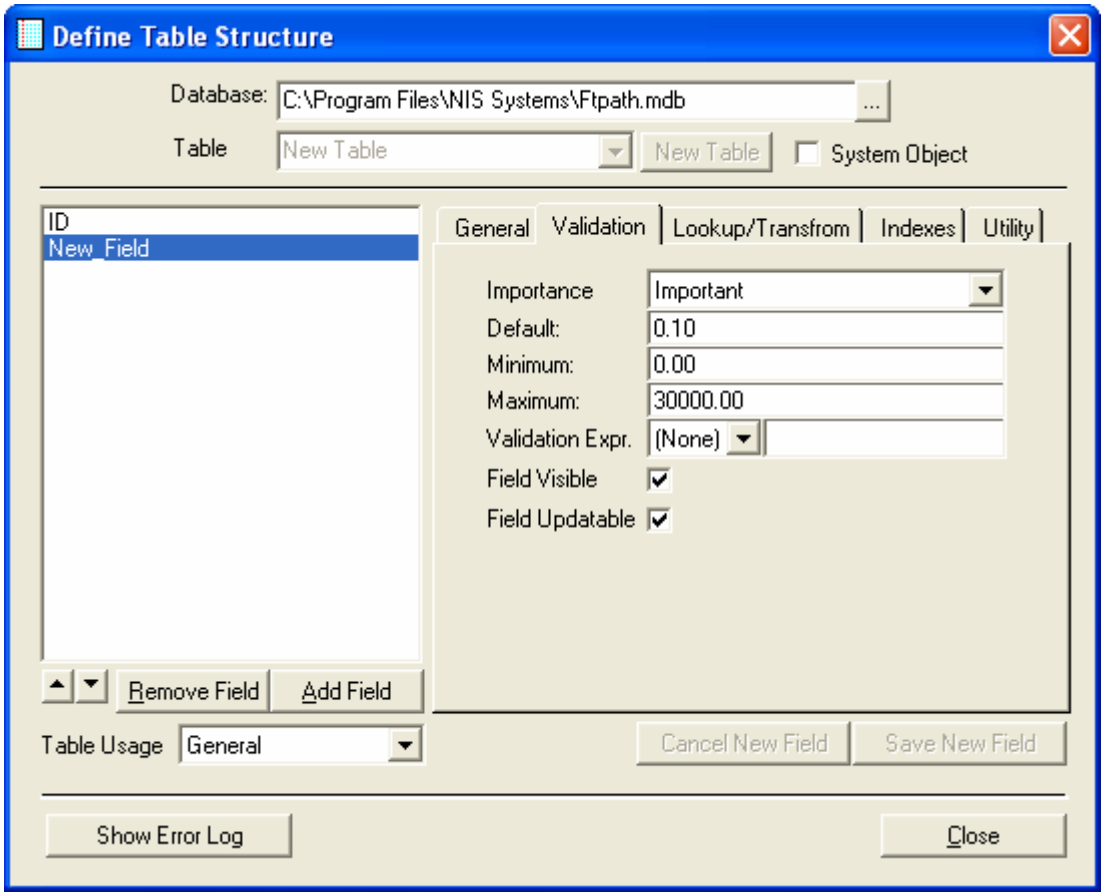
Validation function available allows to check all data in a table and prepare a list of those records which fail the validation.

Data validation Rule is defined under **Validation** tab in 'Define Table Structure' window.

The parameters used to define validation rules include:

Parameter	Available Option	Description
Importance		Importance of data in terms of validation requirement (a grouping value)
	General	Does not validate the data field
	Important	Validate data field and flag the data as exceptional data

	Unimportant	Validate data field and replace missing or erroneous data with default value
Default		Default value use to update the data field value for unimportant data
Minimum		Minimum data applicable to the data field
Maximum		Maximum data applicable to the data field
Validation Expression		Expression for validation (such as field value must be in a list)
Field Visible	Yes/No	Option whether to display field in data view
Field Updatable	Yes/No	Option whether to allow to manually modify value in the data field



4.5 Defining Indexes for Sorting Data

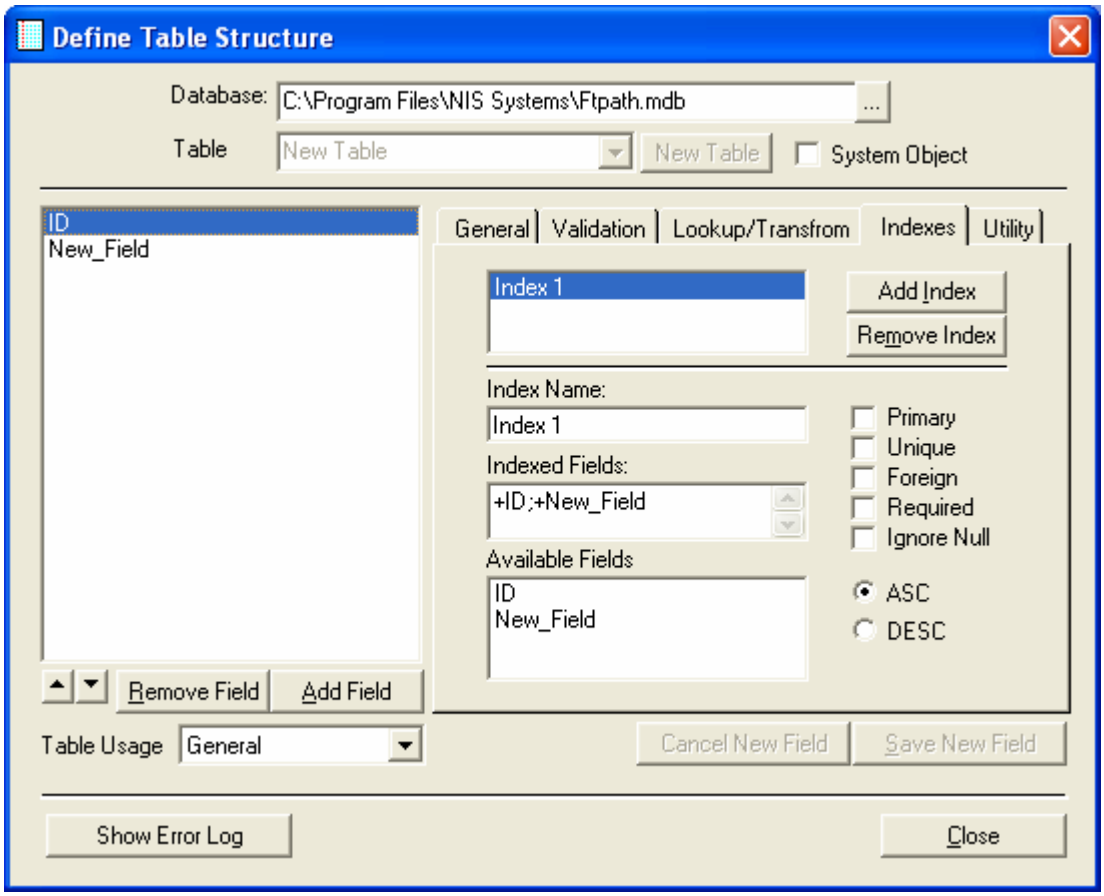
The '**Indexes**' tab includes options to create indexes for sorting the data available in the table. Indexes are used to increase the speed of searches, but also result in larger files. They are not generally required except for sophisticated databases.
To create an index select **Add Index** button. The index name should be defined (without any spaces). The fields to be used for indexing should be selected from the **Available Fields** box. When selected, they appear in the **Indexed Fields** box.

There are several options to use with indexes:

Primary: This is the primary index
Unique: This ensures that the index is completely unique in the system. It is recommended that this be turned ON.
Foreign: Indicates that this is a secondary key linked to another table
Required: Data must be present. This will force the records to be populated.

Ignore Null: Ignores null values when creating an index. It is recommended that this be turned ON.

Once the fields have been selected and the index options defined select **Save New Index** to create the index.



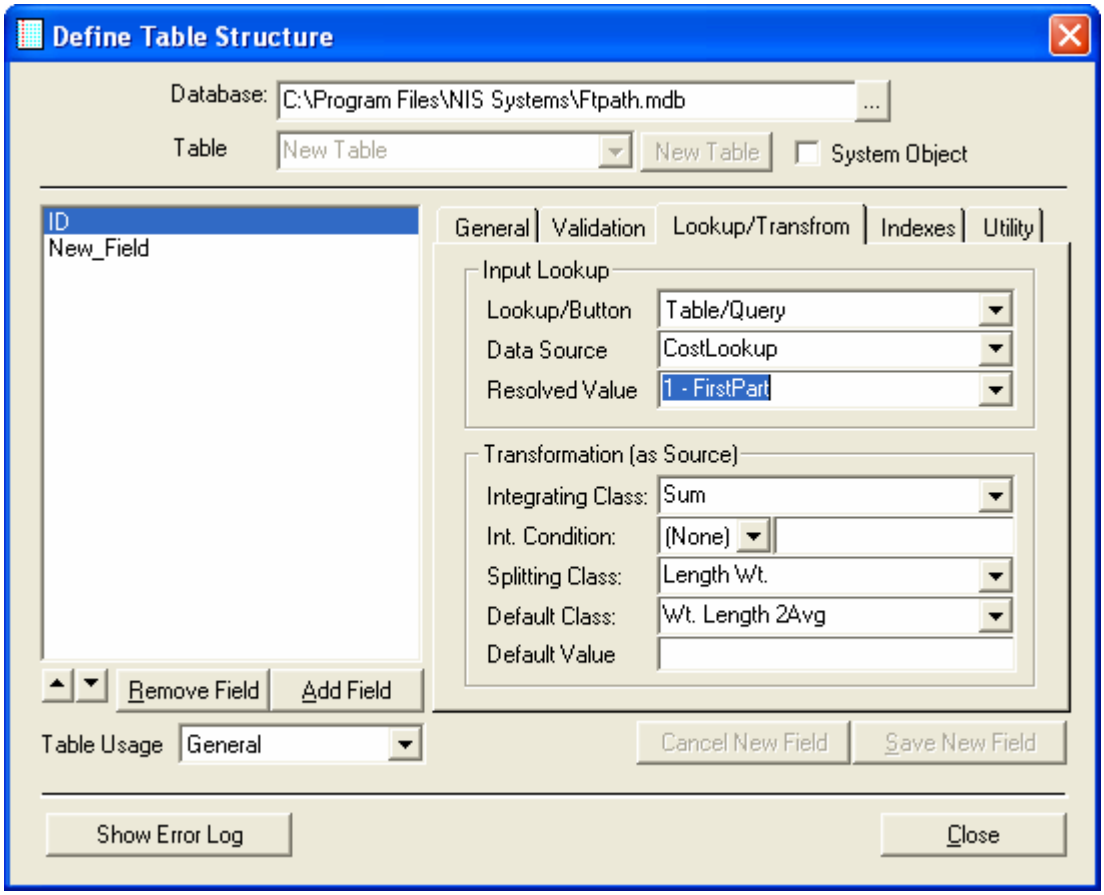
4.6 Lookup / Transformation

The Lookup/Transformation tab is used to define lookup tables and the way in which data are transformed (summarised) from individual sections to larger or smaller sections.

Input Lookup option is used to link the field in the table to another data source.

Parameter	Available Option	Description
Lookup / Button		Allow to select an option for data lookup or button operation for data entry. Options to be displayed in **Data Source** depends on the selection
	(None)	No operation or delete old definition
	Value List	Allows to create a Value List in Data Source to be displayed for selection
	Table/Query	Select a Data Table or Query from which data will be used in Data sources
	Button Action	Assign data field with one of the actions below : Browse for File Browse or Open File Browse and Open File Open File

		Open With...
		Load Picture
		Show Font
		Show Color
		Display Video
		Display Image
		Display Graph
		Display Table Data
		Display Query Data
		Display Form
	All Tables	Display all tables as list of selection
	All Queries	Display all queries as list of selection
	All Tables/Queries	Display all tables and queries as list of selection
	All Fields from Tables	Display all fields from a table (or query) named in a field of this table defined in **Data Source**
	All Objects of Type	Display all objects of a certain type named in a field of this table defined in **Data Source**
	Sectioning Definition	Display sectioning definition function (for Sectioning object only)
	Open Object of Type	Open a object (named in a field of this table defined in **Object Name**) of type (named in a field of this table defined in **Object Type**)
	Filtering Definition	Open filter building function from a data source named in a field of this table defined in **Data Source**
	Enumeration Definition	Open Enumeration Definition function for a field (named in a field of this table defined in Field Name) from a data source (named in a field of this table defined in **Object Name**)
Data Source		Allows to select a table or Action or Object based on the selected **Lookup/Button** option
Resolved Value		Define which part of the value displayed to be used
Object Name **Object Type** **Field Name**		Dynamically displayed caption for defining different types of button operations



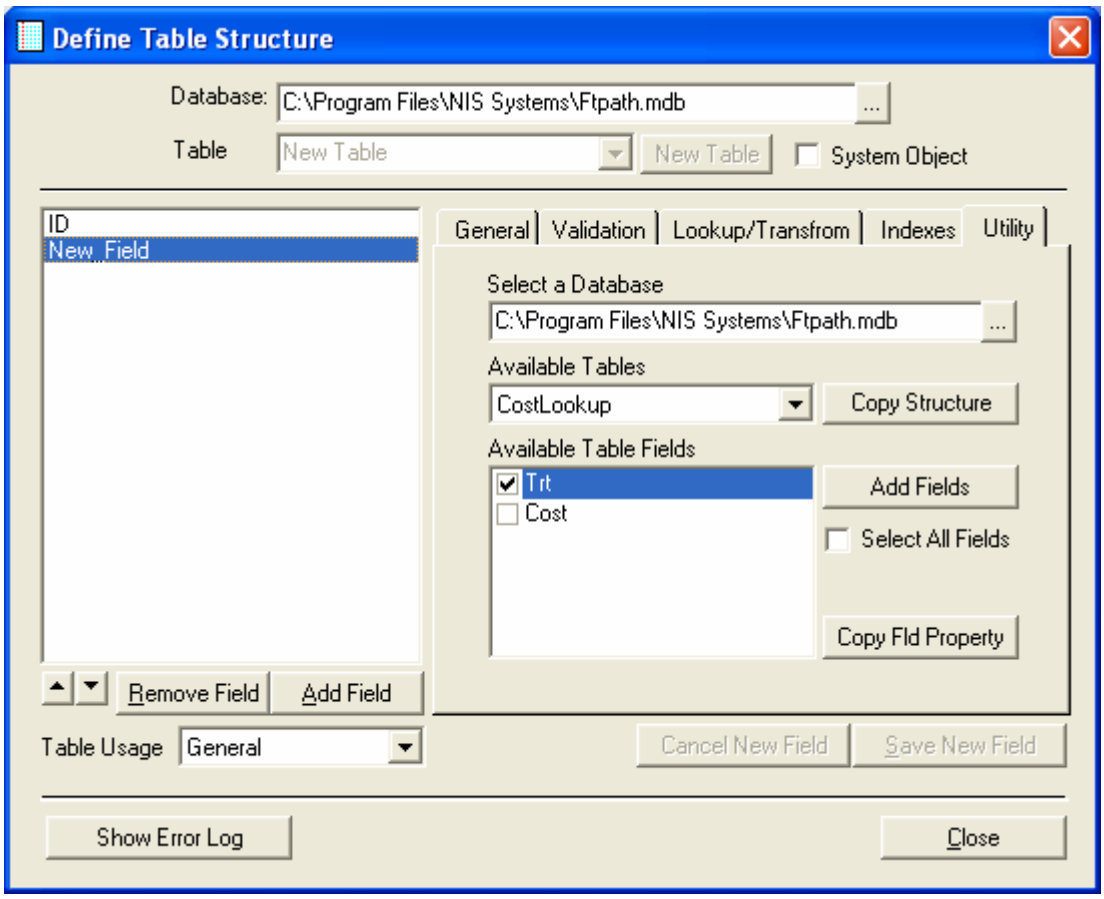
Transformation definitions included for the mathematical operation to be performed while summarising the data to larger or smaller sections are as follows. The options available for each field are dependent on field data types.

Parameter	Available Option	Description
Integrating Class		The way in which data from individual small sections will be combined to larger sections
	Avg	Average
	Count	Count
	First	First
	Last	Last
	Max	Maximum Value
	Min	Minimum Value
	Most Length	Most Length
	StDev	Standard Deviation
	Sum	Sum
	Wt Length Avg	Weighted Length Average
	Length Sum	Sum of Length
	Wt Length Sum	Weighted Length Sum
Int. Condition		Data aggregation from a list of operations and data definition for comparison
Splitting Class		The way in which data from larger sections are split into smaller sections
	Same Value	Same value
	Length Wt.	Value split based on Length
Default Class		When there are null or missing values how these will be handled
	Default	Use value defined in **Default Value**
	Avg	Use average for the Link
	2Avg	Use average of 2 adjacent sections

	Zero	Apply Zero Value
	Null	Apply Null Value
	Previous	Use value of previous section
	Next	Use value of next section
	Wt. Length 2Avg	Use length weighted average of 2 adjacent sections, ie (v1*s1+v2*s2)/(s1+s2)
	2 Length Wt	Use average of rate or the two adjacent sections, ie (v1/s1+v2/s2)*(s1+s2)/2
Default Value		Value if defined will be applied as default value during transformation if value is missing. The default value is applied only if Default value is selected as Default Class

4.7 Table Utility

The utility tab is used to import structure of a table. The data source could be Access, Excel, Text or dBase file. User needs to first **Select a Database** available. Then he can select a **Available Table** in the database.



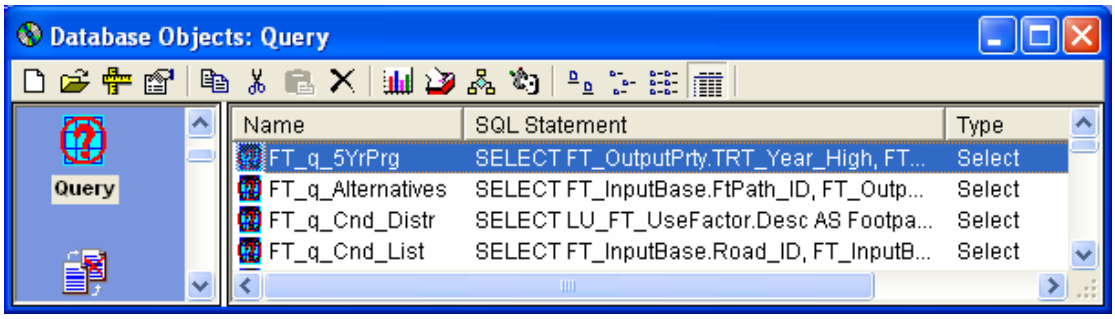
Various option for selection of the fields are available including :

Copy Structure: Copy the design of whole table
Available Table Fields: Allows to select individual fields available in the source table
Select All Fields: This check box will select all the fields available in the source table
Copy Fld Property: Applies the field property of the highlighted field of the source table to highlighted field of the current table

5 WORKING WITH QUERIES

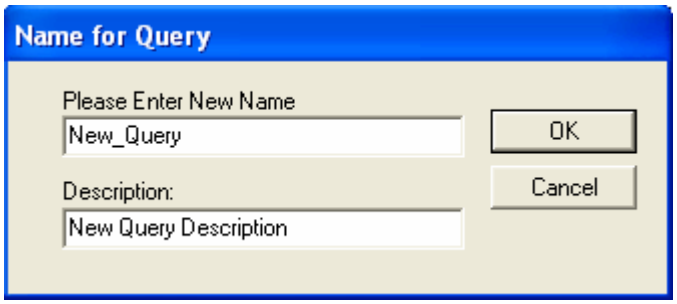
Queries are extensively used by NODEM. It can be included in most of the objects including Report, Chart, Variables, Models, Batches, Analysis etc. A very user friendly query builder wizard is available to build the query.

The list of all the available queries are included in **Data Objects: Query** window.



5.1 Create Query

A new **Query** can be created by clicking on  button in the top menu in the **Data Objects: Query** window. **Name for Query** window allows to define the name and description of the table.

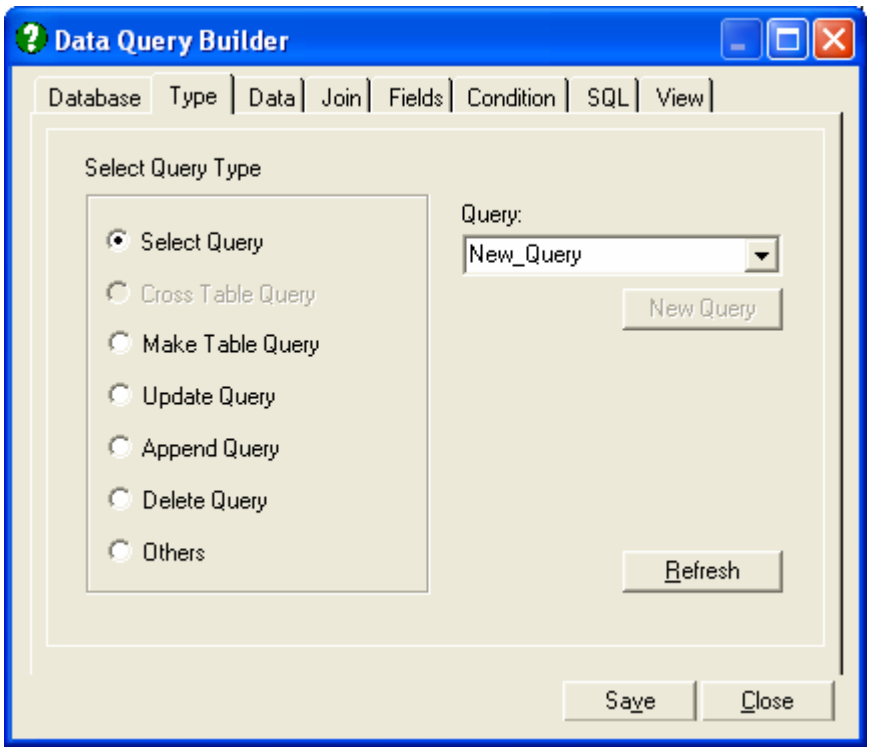


5.2 Design Query

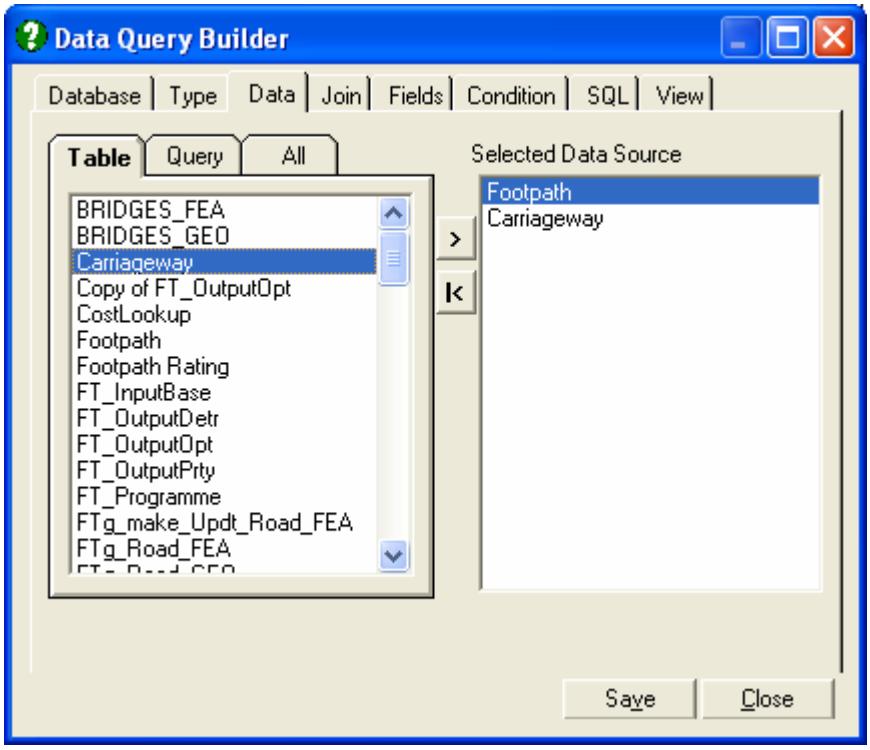
Data Query Builder allows to design different types of query including:

Select Query
Make Table Query
Update Query
Append Query
Delete Query

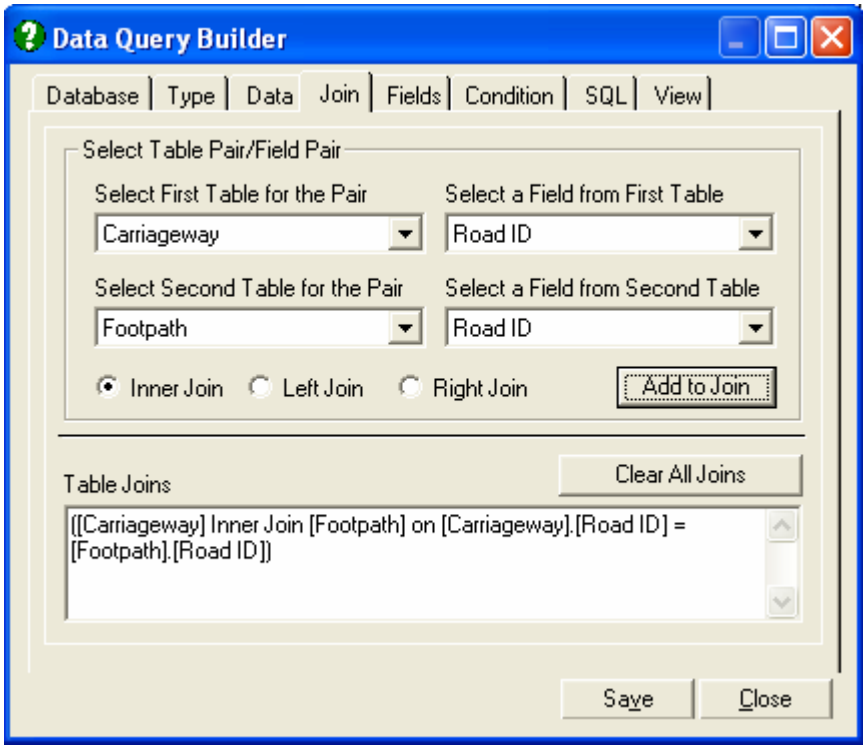
For cross table and other complex queries user can directly type sql statement in the **SQL** tab in **Data Objects: Query** window.



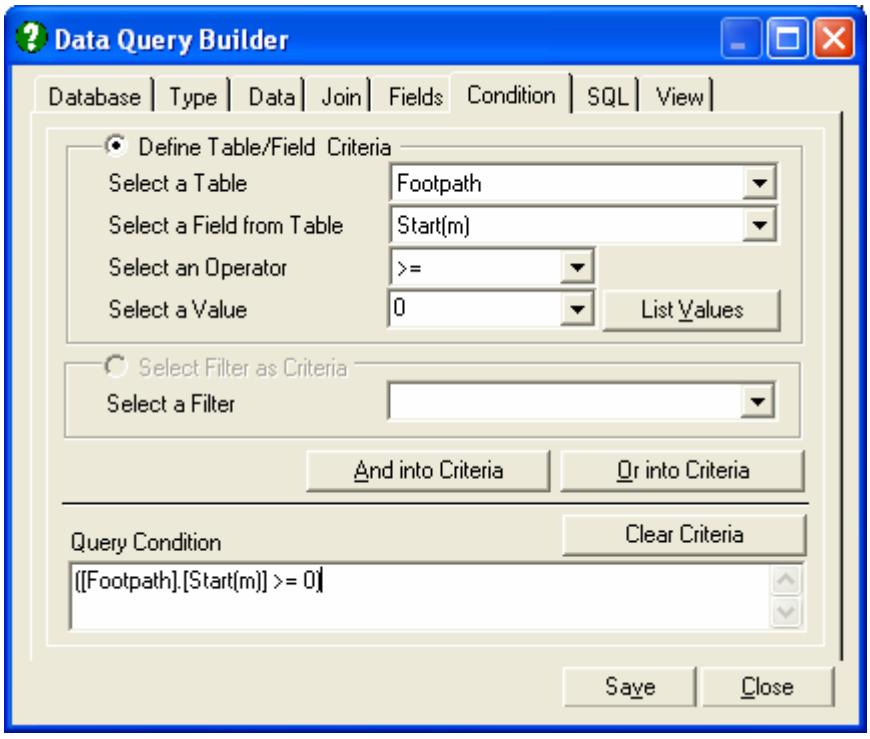
Fields tab allows to select data fields (to be included in queries) from the existing data tables and queries.



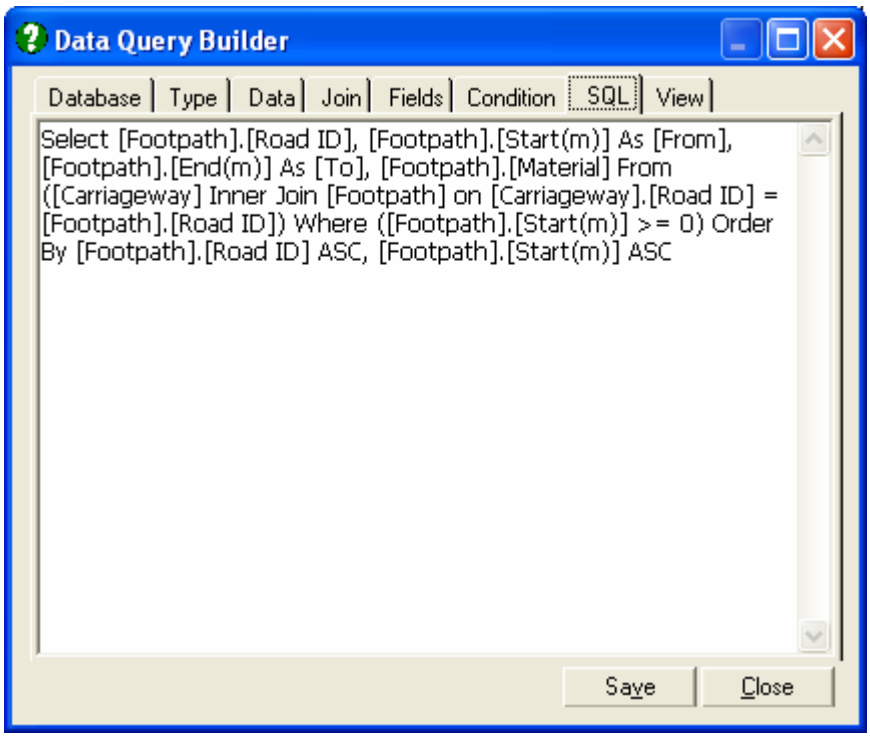
Join tab allows to build relationship between the tables, if more than one tables are used in the query.



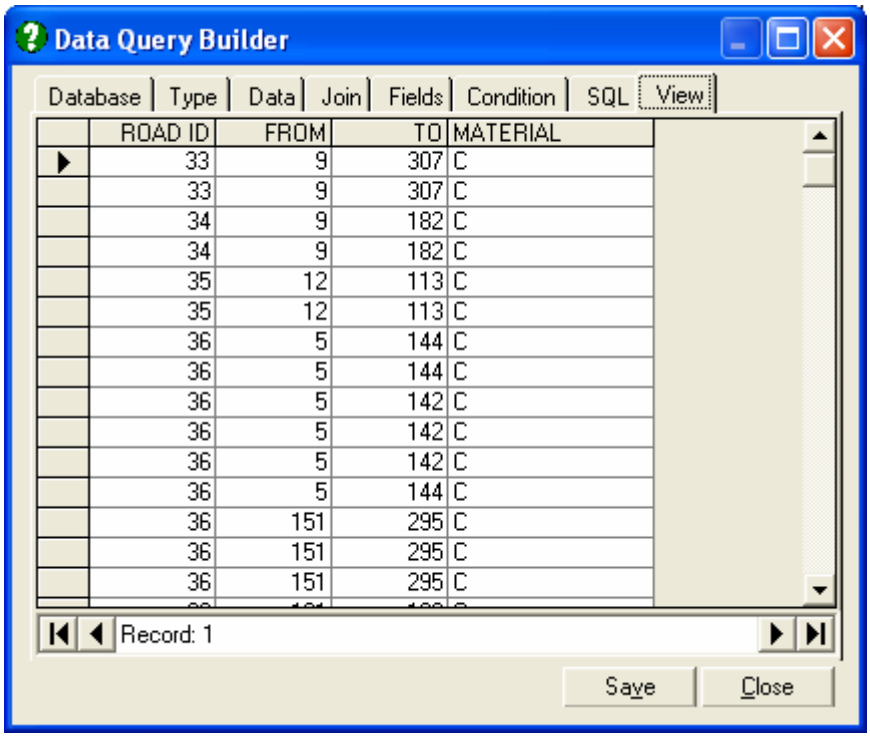
Condition tab allows to build filter condition in the query, if required.



SQL tab allows to do any modifications to the SQL statement if required. Please note that any modifications undertaken in the SQL tab may not be retained, as the statement will be rebuilt if modification in any of the tabs except **View** is undertaken.



View tab allows to view the results of the query



5.3 Use Query

To view an existing query by highlighting the query and click button.

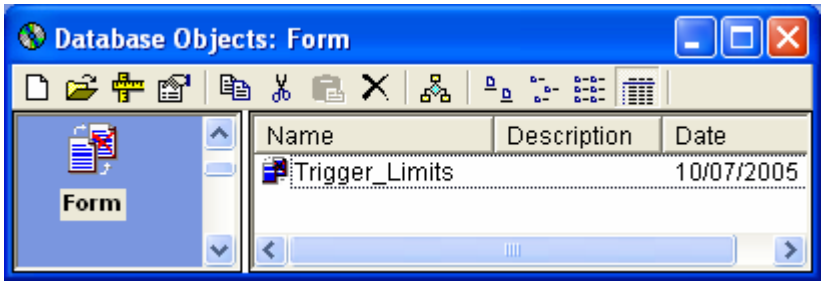
To modify a query by highlighting the query and then click button. The query builder will be displayed for required modification.

The existing query can be included in most of the objects including Model, Batches, Analysis, Reports, Charts etc. Action query (such as append, delete and update etc) can only be run by user with certain privileges and access modes.

6 WORKING WITH FORMS

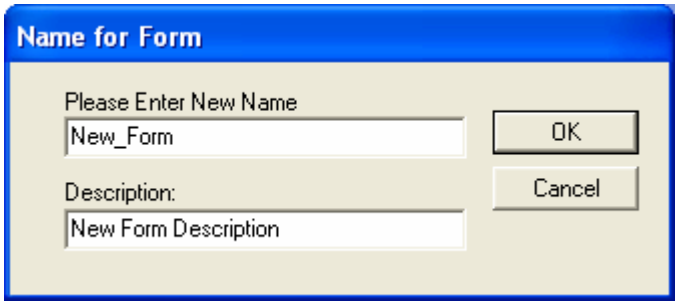
A versatile and easily customisable data form designing tools is available in NODEM. The Form can be used to enter the data in to the database or to provide parameters at the time of running various functionality available in the system.

A list of all the Forms can be viewed in **Database Objects: Forms** window.



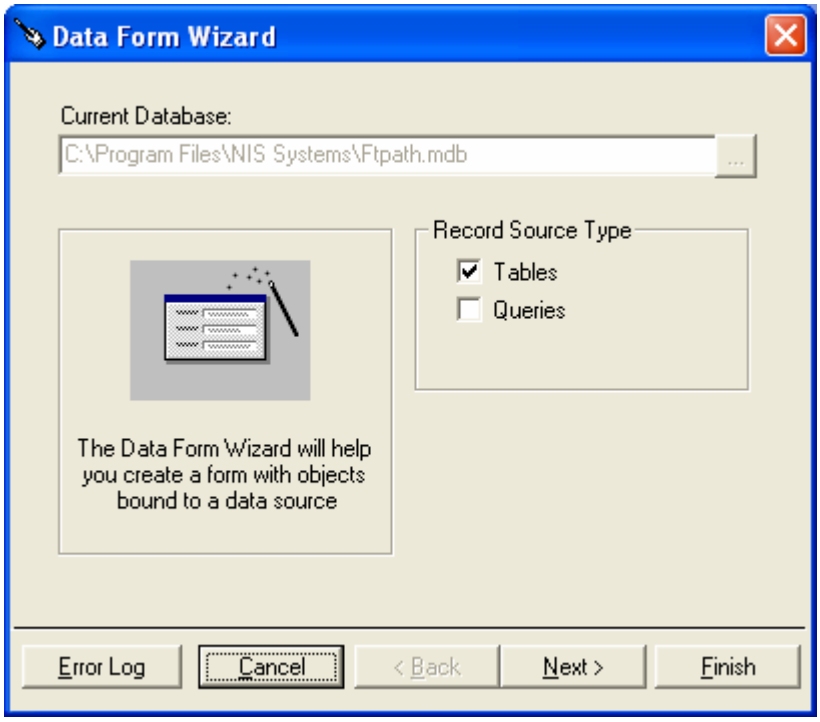
6.1 Create Form

A new **Form** can be created by clicking on  button in the top menu in the **Data Objects: Form** window. **Name for Form** window allows to define the name and description of the Form. Click **OK** to save the Name for Form.

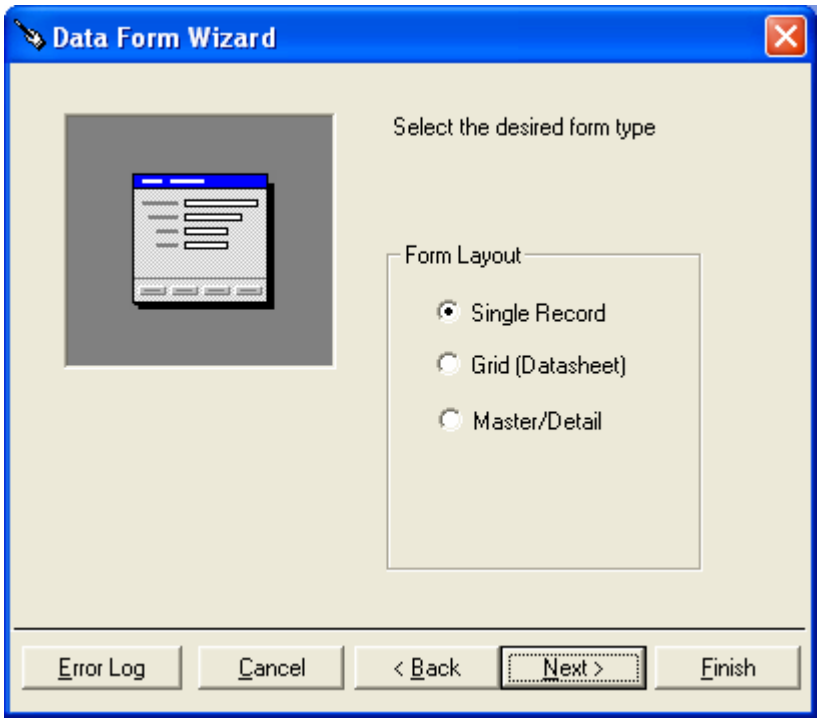


6.2 Design Form

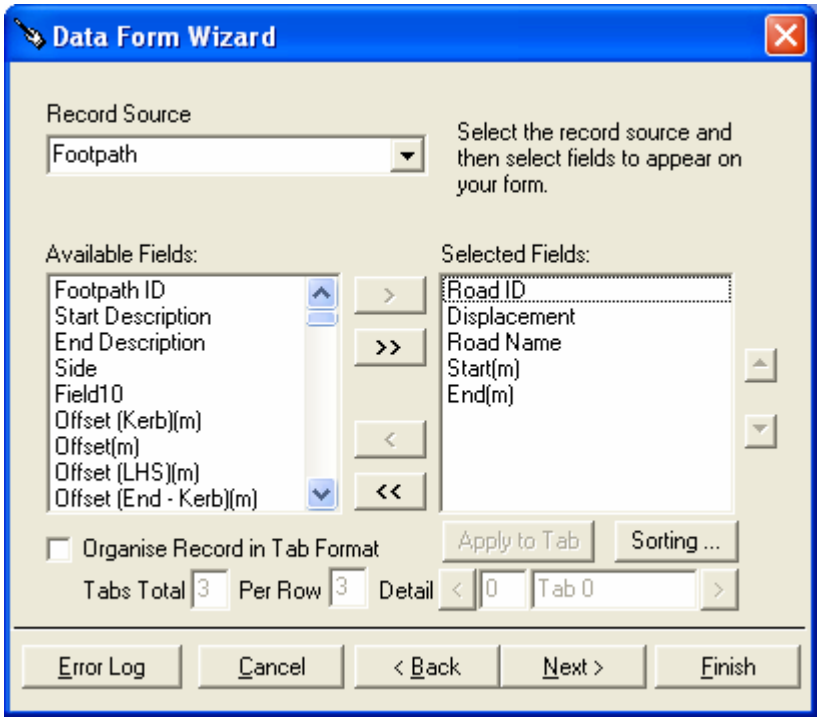
A wizard is activated once the Name for Form is saved. Table / Query can be selected as option for the data source to be used by the Form.



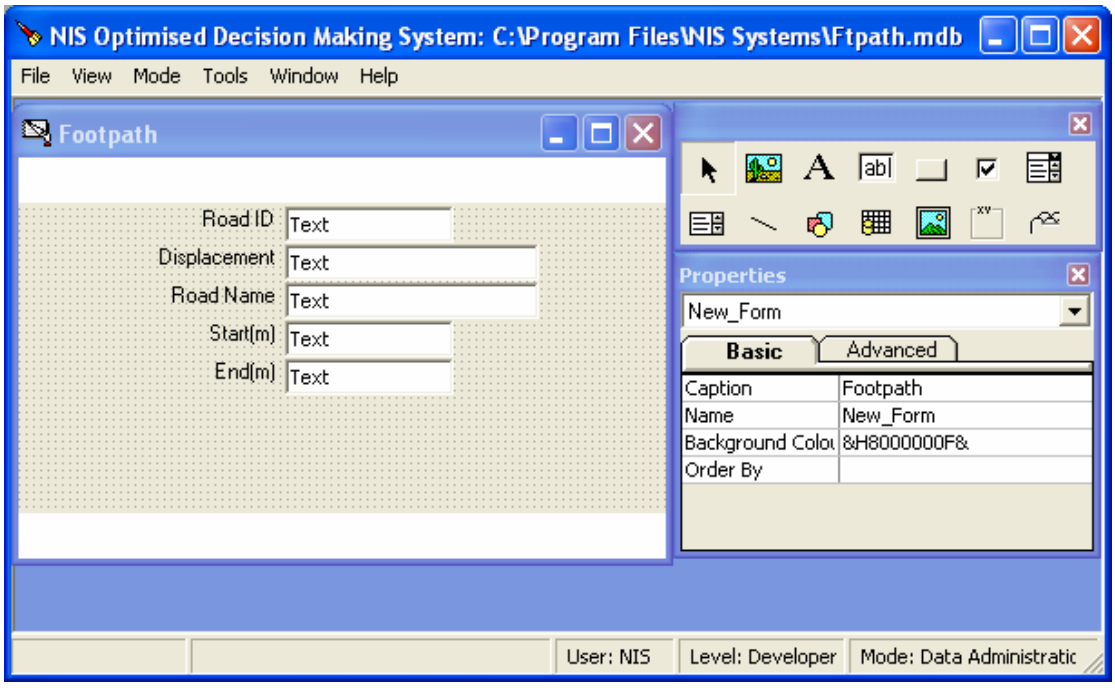
Clicking Next will open the window allowing to select the Form Layout.



Clicking **Next** will open the window for selecting data items to be included in the Form.



Clicking **Next** will generate a Form and open it in the Designer window view. The locations, formats or captions of the generated Form can be easily modified using available functionalities in the Designer. Closing the Form will automatically save the design of the Form.



6.3 Use Form

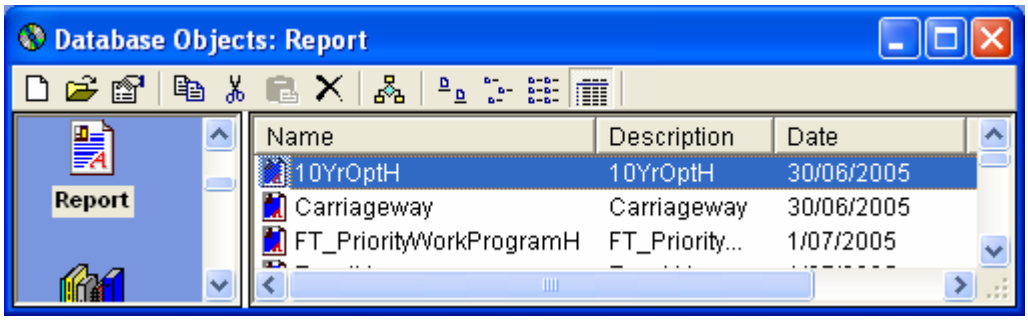
An existing Form can be opened by highlighting the form and then click button.
For modification of a Form, user may need to select/highlight the query and then click button. The Form will be opened in Designer view allowing required modifications.

The Form can be included in most of other operational objects including Model, Batches, Analysis etc. for data operation and manipulation.

7 WORKING WITH REPORTS

The reporting functionality in NODEM is fully integrated with MS Excel Chart functions. The Report Designer allows to extract the data from Table or Query and build a Report template using MS Excel components.

The Report template once saved will be available in **Database Objects: Report** window.



7.1 Create New Report Template

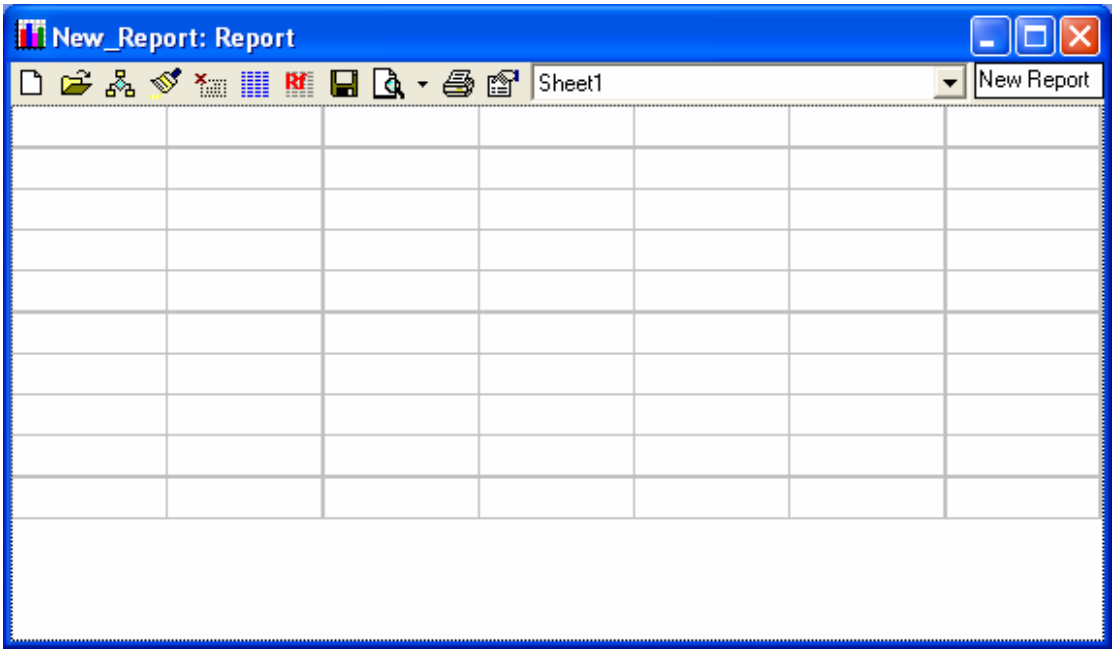
A new **Report** template name can be defined by clicking on button in the top menu in the **Data Objects: Report** window. **Name for Report** window allows to define the name and description of the Report.

Clicking **OK** will save the **Name** and **Description** of the Report Template.

7.2 Design Report Template

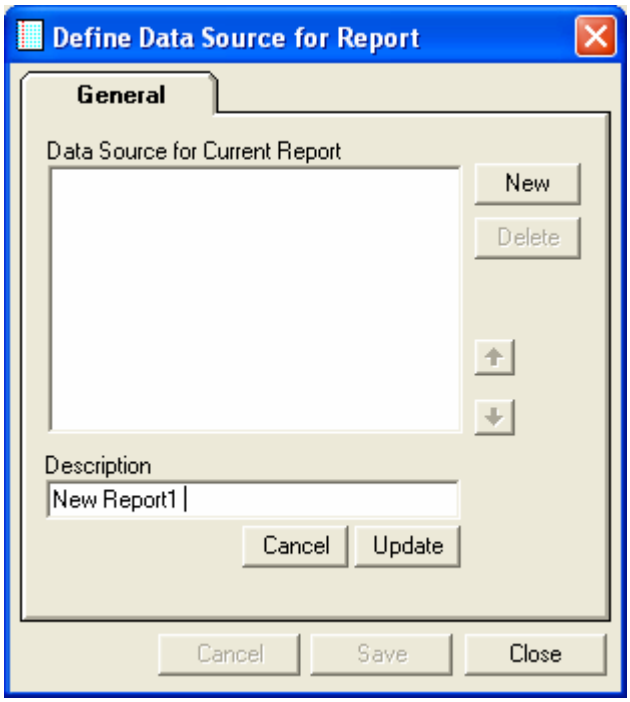
The Report designer window can be opened by pressing.

A newly added report needs to start with a new workbook. To create a new workbook for the report, click on button. A workbook with a single worksheet will be created.

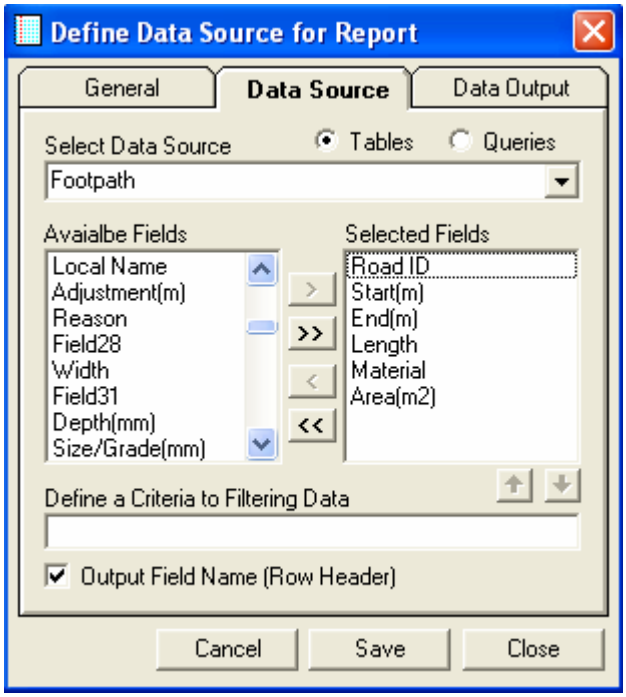


The data source to be extracted to the sheet can be defined by clicking on  button. Define **Data Source for Report template** window will be opened.

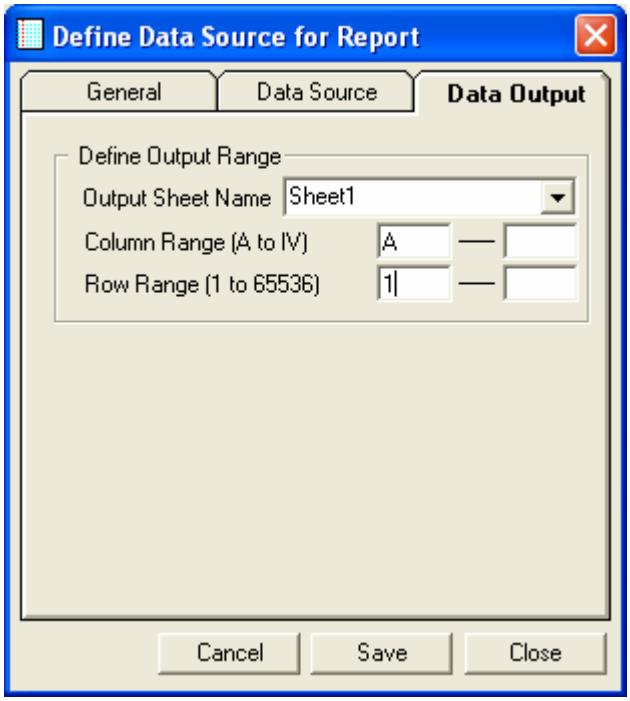
Need to click **New** to define a new data source and then enter name in **Description** box. Pressing **Update** will save the name.



The **Data Source** tab allows to select fields, available in existing Tables and Queries, to be included in the Report template.



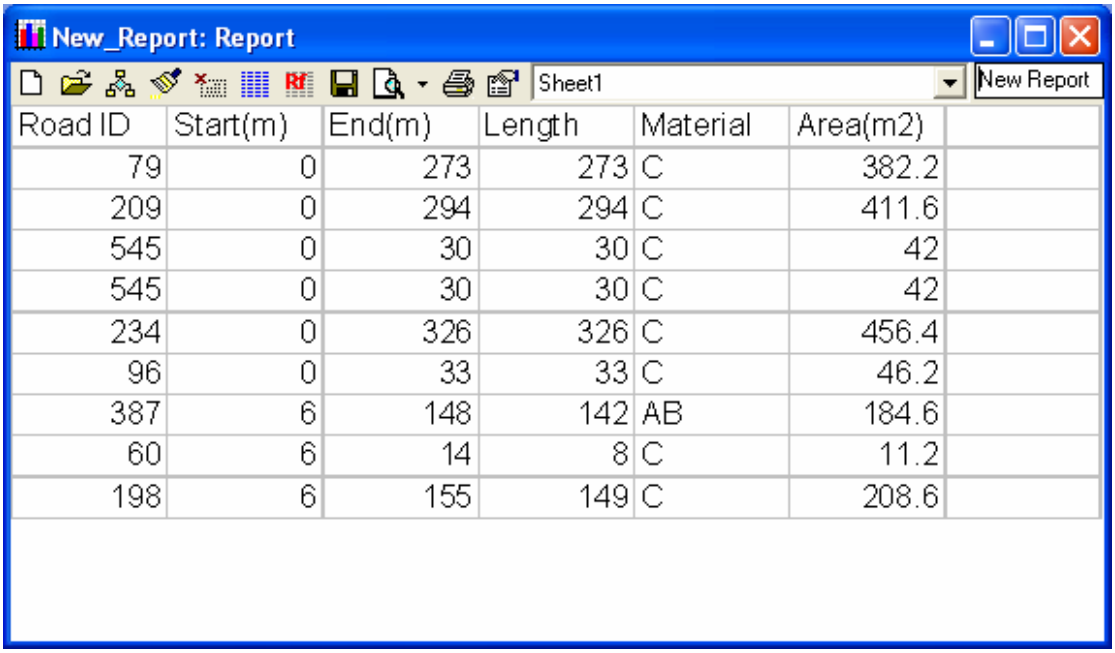
The **Data Output** tab allows to select location in the worksheet where the data are to be extracted.



Click **Save** and **Close** to save and close the **Define Data Source for Report** window.

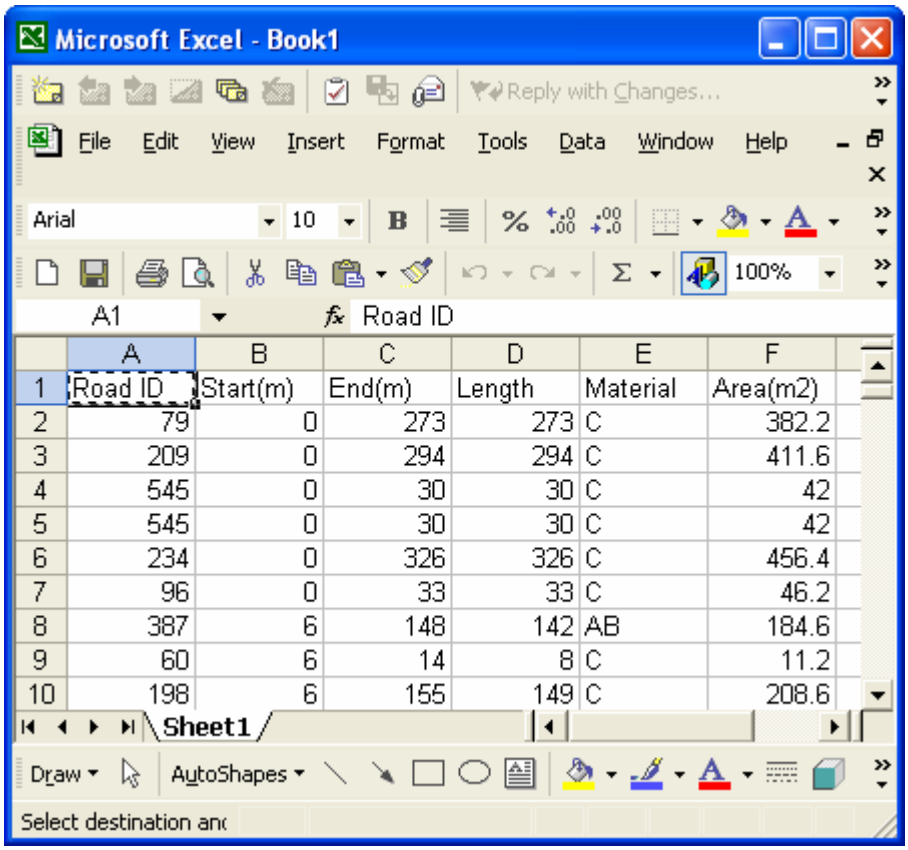
A report can have a list of data sources, but each data source contains one data source (table or query) and one data output definition.

Refresh the data display by clicking on Refresh button. Upon a change in the data source, select the button will reflect the data change in report.

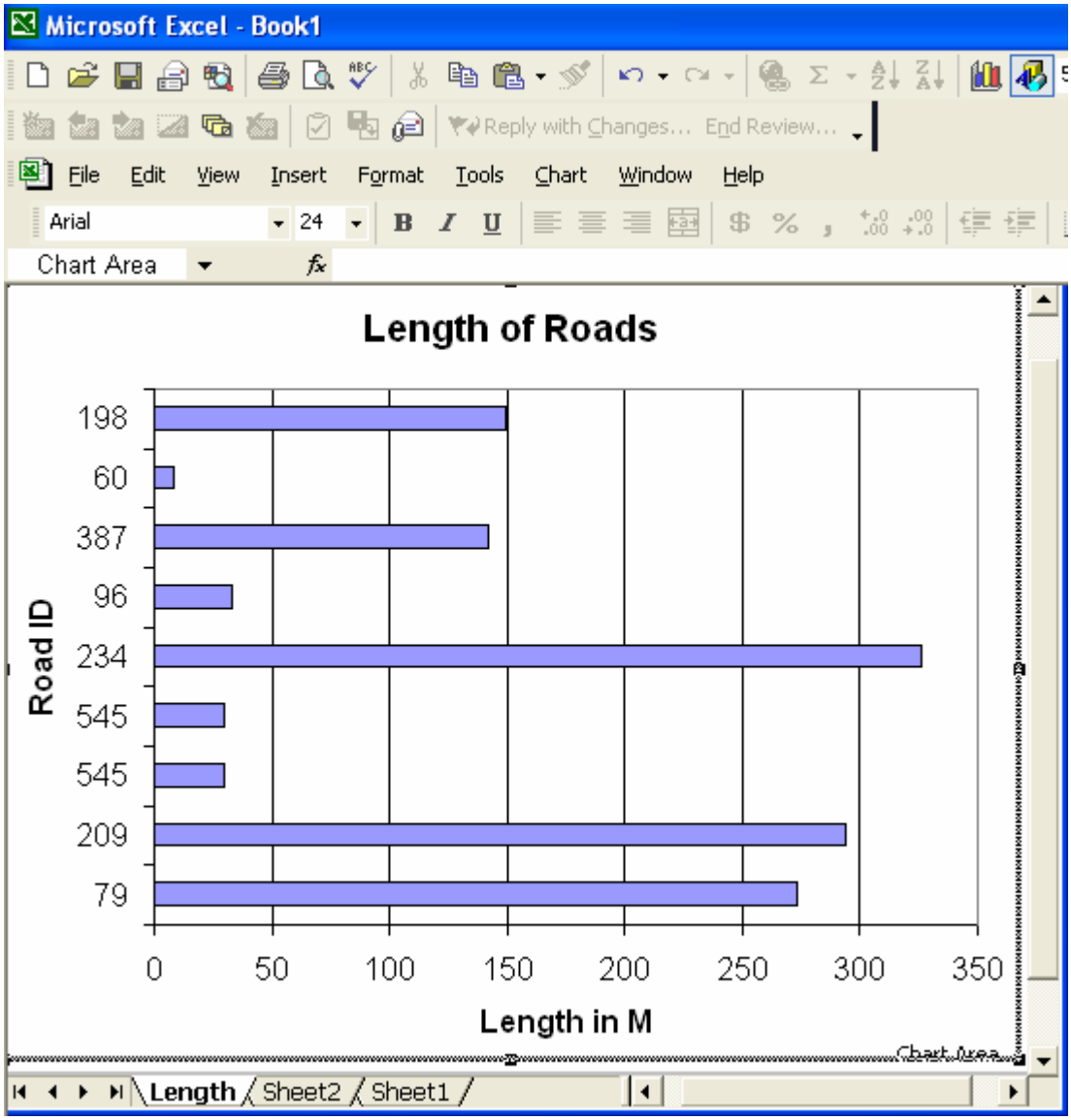


Road ID	Start(m)	End(m)	Length	Material	Area(m2)
79	0	273	273	C	382.2
209	0	294	294	C	411.6
545	0	30	30	C	42
545	0	30	30	C	42
234	0	326	326	C	456.4
96	0	33	33	C	46.2
387	6	148	142	AB	184.6
60	6	14	8	C	11.2
198	6	155	149	C	208.6

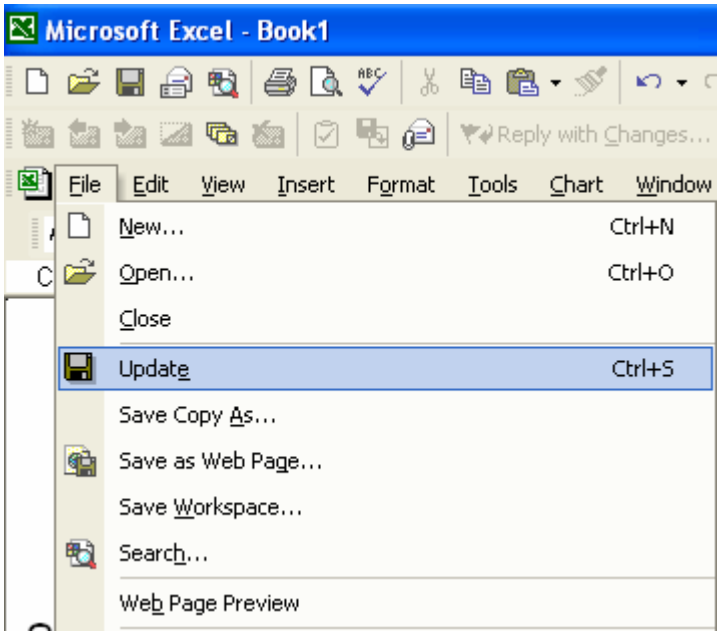
Report can be designed locally by pressing or in Excel by pressing.



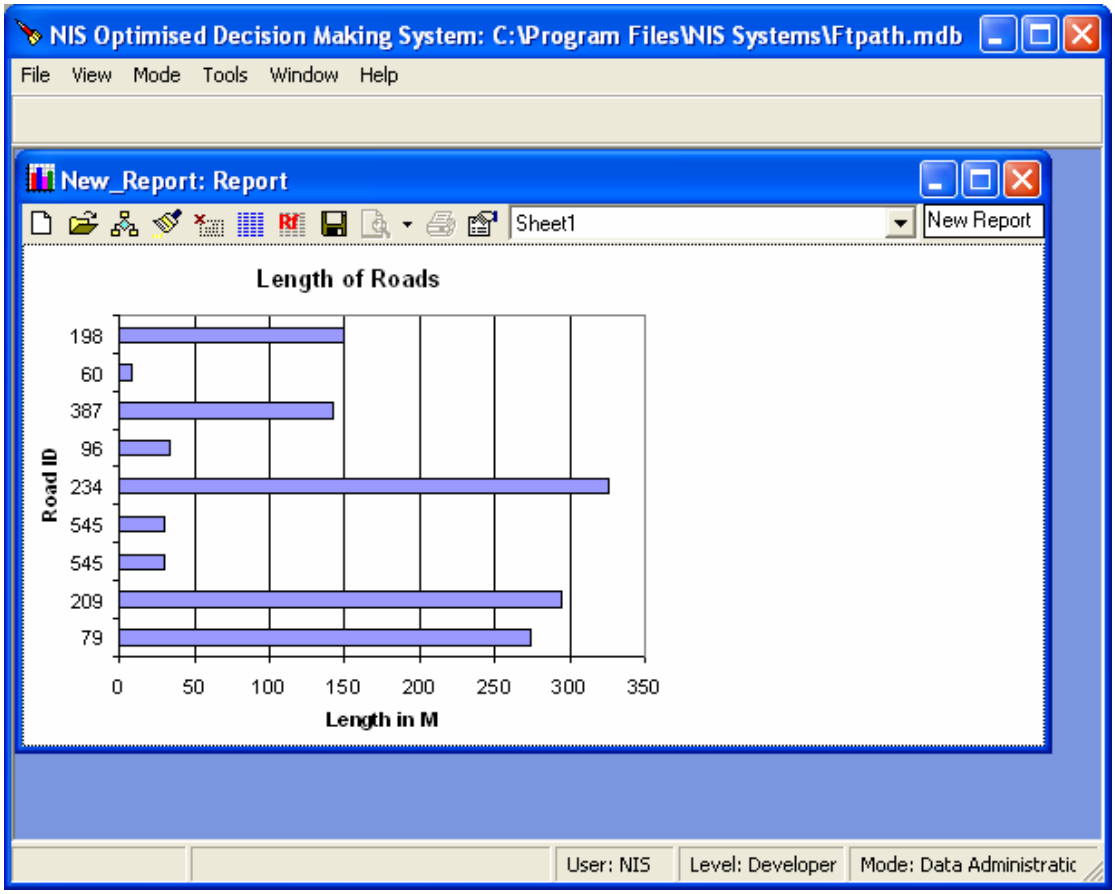
	A	B	C	D	E	F
1	Road ID	Start(m)	End(m)	Length	Material	Area(m2)
2	79	0	273	273	C	382.2
3	209	0	294	294	C	411.6
4	545	0	30	30	C	42
5	545	0	30	30	C	42
6	234	0	326	326	C	456.4
7	96	0	33	33	C	46.2
8	387	6	148	142	AB	184.6
9	60	6	14	8	C	11.2
10	198	6	155	149	C	208.6



Save the report template by selecting **File /Update.**



Close the MS Excel. This report template is now saved in NODEM database.



7.3 Use Report Template

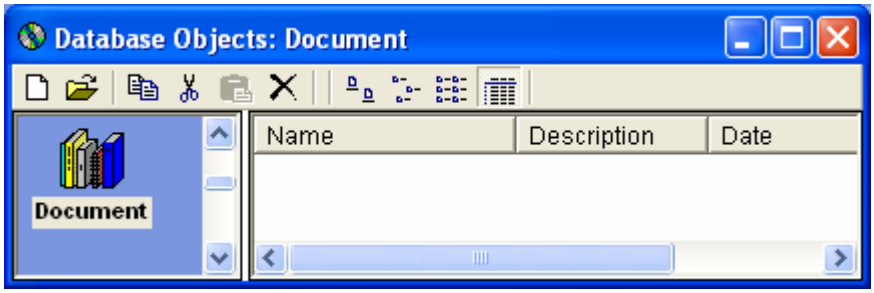
To use a Report Template, select the report in the **Database Objects: Report** window and press  . This will open the Report in a preview mode. If the report needs to be printed or data to be extracted, open the Report in MS Excel format by pressing  .

Generally Reports are opened from Navigator or Analysis. The functionality available in Subsystem and Analysis objects allows to include the Report Templates.

8 WORKING WITH USER DOCUMENTS

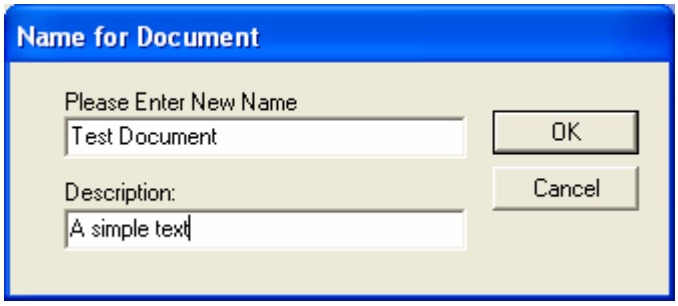
User documents refer to working notes the user write down in the working process or period using NODEM system. It is not a document management system used to manage and control access to different types of documents (which can be easily implemented through NODEM)

All available user document set can be viewed in **Database Objects: Document** window.

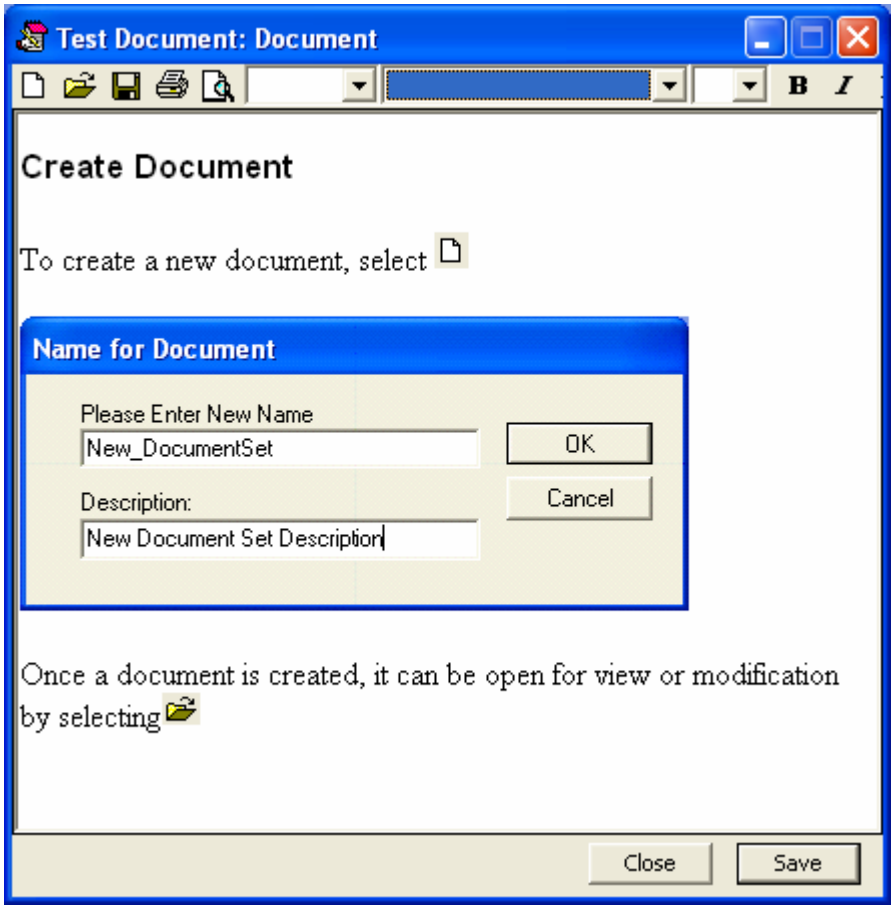


8.1 Create Document

To create a new document, select 



Once a document is created, it can be open for view or modification by selecting  . User document is more useful for information tracking rather than editing. User can use other software such as **Word** to do the documents and then copy through and save inside the database.



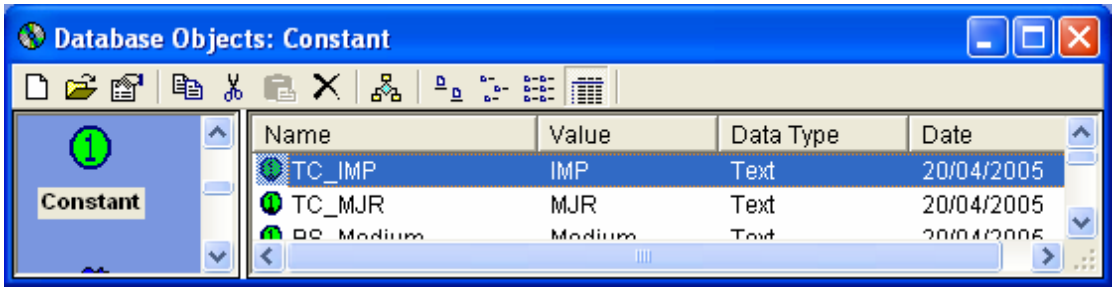
8.2 Use User Document

User document can be included in other operations for display and modification.

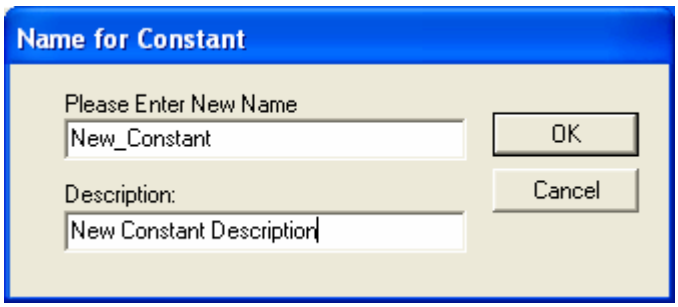
9 WORKING WITH CONSTANTS

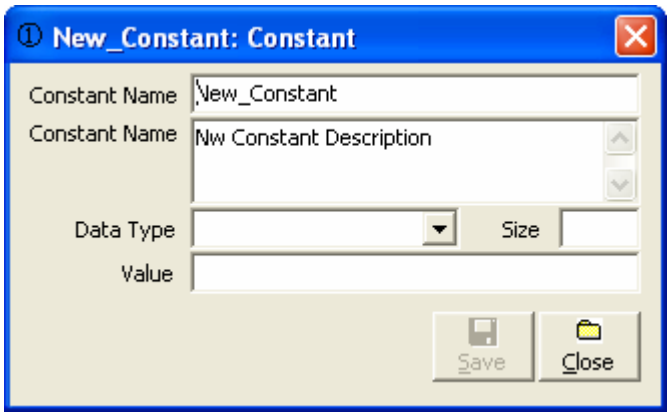
Constants in NODEM is used when certain parameter can be given global default value for a parameter.

All the available constants are kept in **Database Objects: Constant** window.



9.1 Create Constant



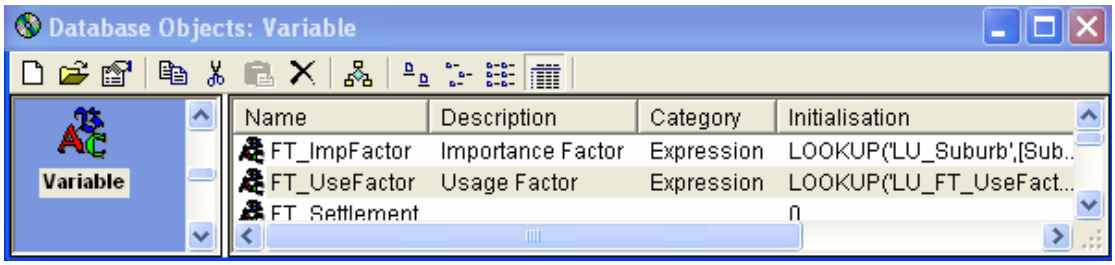


Click Save to save the constant once defined.

9.2 Use Constant

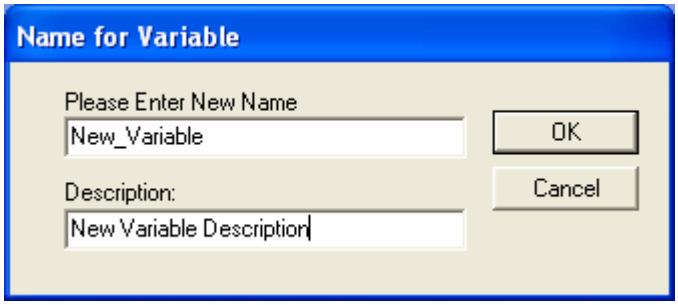
Constant can be used as part of Variable and Model definitions.

10 WORKING WITH VARIABLES

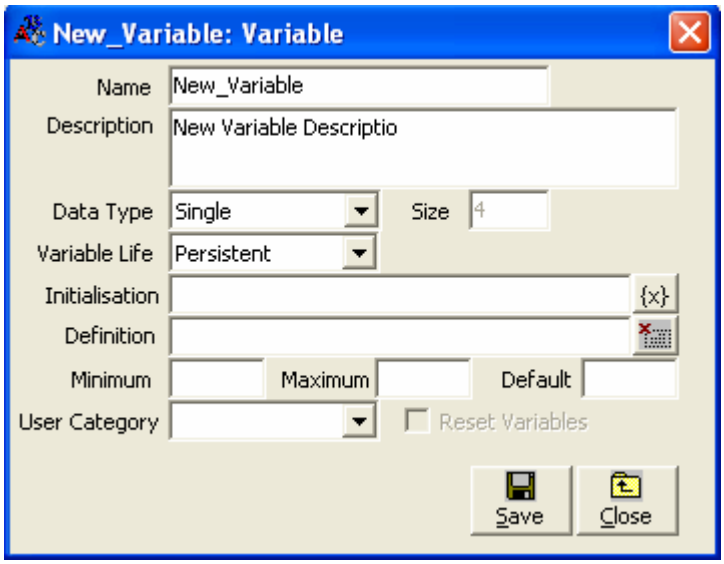


10.1 Create Variable

A new **Variable** can be defined by clicking on  button in the top menu in the **Data Objects: Variable** window. **Name for Variable** window allows to define the name and description of the **Variable.**



Clicking **OK** after entering the Name and Description will open the Variable Definition window.



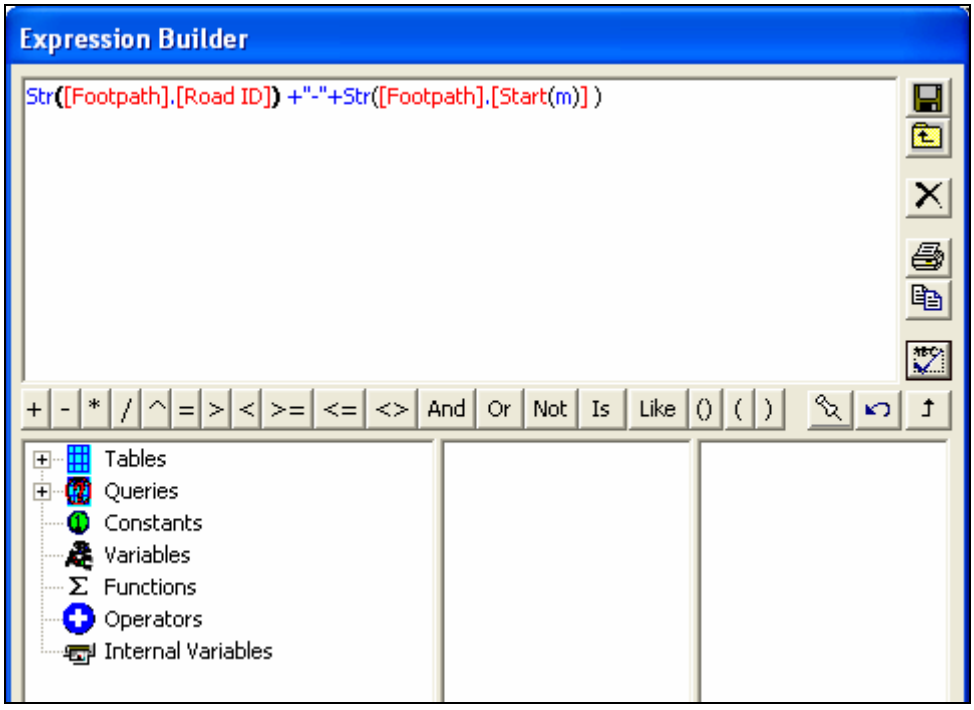
The following parameters need to be defined for each variable:

Data Type		Variable data type
Variable Life	Persistent	Value or the variable will be saved as output
	Volatile	Value of the variable will exist in memory only

Initialisation		Expression used by Initialise command
Definition		Expression used by Definition command
Minimum		Minimum value possible. If the data generated for the variable is less than this minimum value, this minimum value will be applied.
Maximum		Maximum value possible. If the data generated for the variable is less than this maximum value, this maximum value will be applied.
Default		Default value will be applied when the value generated by this variable is null.
Category		User defined category of variables for grouping (view) only. The possible value can be Expression, Filter, Input, Intervention, Treatment etc

10.2 Define Variable Expressions

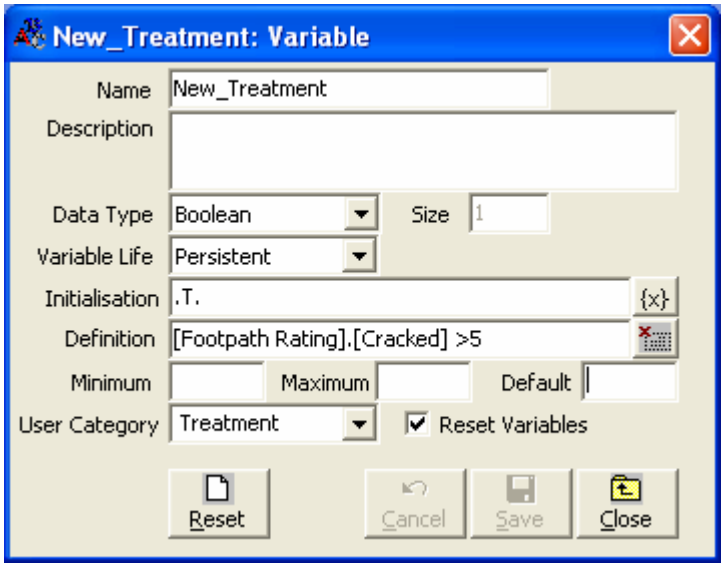
Expression Builder is available for both **Initialisation** and **Definition** Variables.



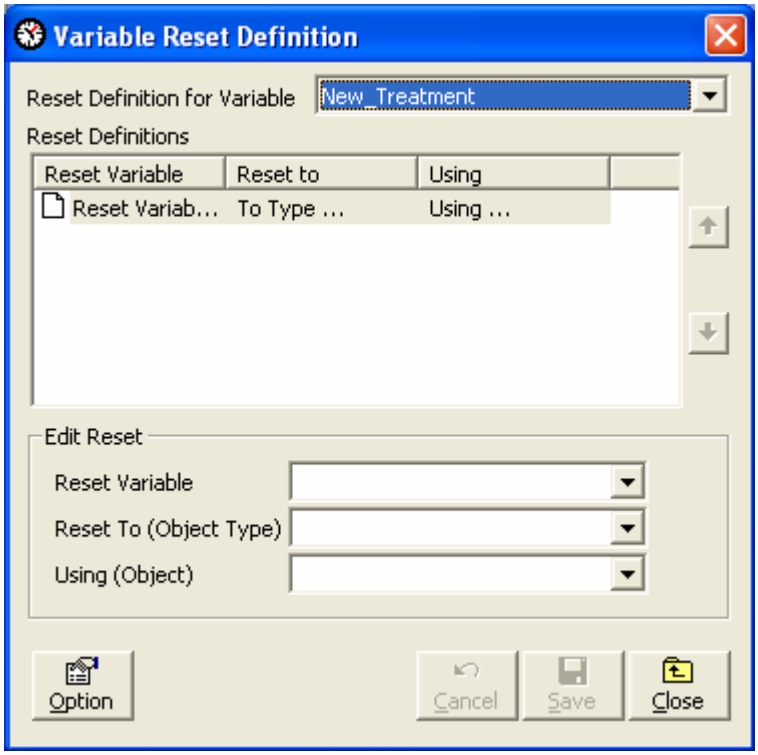
10.3 Define Resets for Reset Variables

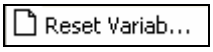
Resets are required when the data for a given asset component has to be updated if the condition defined by the expression is true. A typical example is reseting the condition of pavement surface is when a maintenance treatment is applied.

Reset button is enabled, when **Data Type** is 'Boolean' and **Reset Variables** option is checked.



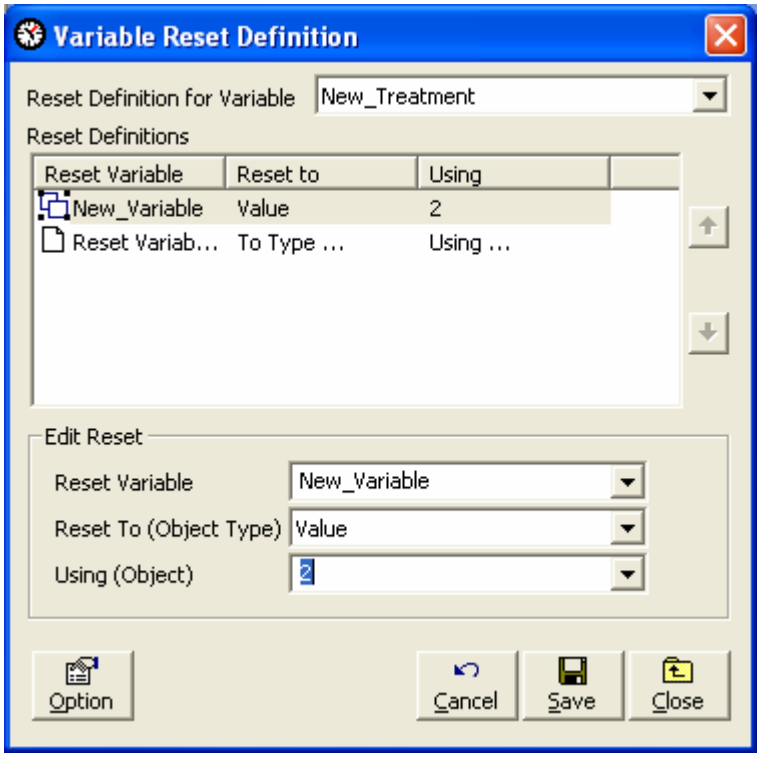
Clicking the Reset button will open the Variable Reset Definition window.



To add a Reset Definition click on  at the right hand side of 

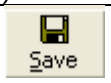
. It will change to  . A new line with

will appear.



Need to select the variable to be reset using drop down menu of **Reset Variable** combo box. The optional object types available for Reset in Reset To (Object Type) combo box include:

Reset To (Object Type)	**Description**
Variable	Allows to select from a Variable (defined in the database) from **Using (Object)** box
Variable (Expression)	Allows to select from a Variable defined as **Expression** in **Category** (defined in the database) from **Using (Object)** box
Constant	Allows to select from Constants (defined in the database) from **Using (Object)** box
Value	Allows to define absolute value as Reset in **Using (Object)** box
Expression	Allows to define expression as Reset in **Using (Object)** box
To Type	Default value to be replaced (enetered)

More than one variable can be defined for Reset. Click 

to save the Reset Definitions.

10.4 Use Variables

Variables are used in Models for data processing.

11 WORKING WITH RUNTIME (DIRECTIVE) VARIABLES

Runtime variables can be used inside models, batches and triggers where they are simply defined in-line with other definitions. Runtime variables are mainly used for gaining user input, confirmation and some preliminary calculations before or during object execution.

11.1 Types of Runtime Variable

Runtime variables can be broadly classified as 3 types: input, output and calculation.

Input Variables
There are certain 12 types of user input variables. But they can be grouped into 5 types as following:

- *Textbox*, to gain single user input. Including InputTextBox, InputFileFolder, InputFileOpen, InputFileSave
- *Combobox*, to gain single input from a list of selections. Including InputComboBox, InputObjectComboBox, InputFieldComboBox
- Listbox: to gain multiple input from a list of selections. Including InputListBox, InputObjectListBox, InputFieldListBox
- Checkbox: to gain multiple choices from a list of selections. Including InputCheckBox only
- Confirmation: to gain confirmation (and a selection from a few choices). Including InputConfirmation only.

Input Calculations
Variables for calculation without saving. Including only 1 type – InputCalculation

Output Calculations
Variables for calculation and output to destination data table. Including only 1 type - OutputVariable

Output variables can only be used inside trigger definition.

11.2 Define Runtime Variable

Runtime variable is in-line definition inside object (model, batch or trigger) etc. Format for defining runtime variables is the same for all types of object, although the head may slightly different. Following are the definition format for runtime variables

ID	Operation	ObjType /ObjectType	ObjName /ObjectName	Param /Parameter	FldName /Parameter2	Open	Comment	Included	InclAtRunT
Order	InputFileFolder	Prompt	Variable	(Empty)	Default	(No action)	comment	True/False	True/False (exp or var)
1	InputFileFolder	Please select the RAMM Unload Folder	Loc	(File Pattern if InputFileOpen/Save)	Default			TRUE	
Order	InputComboBox	Prompt	Variable	(Empty)	Default	(No action)	comment	True/False	True/False (exp or var)
2	*InputComboBox*	*Define Analysis Area*	*Ward*	*Select Ward from qry_Suburb*	*HOBSON*			*TRUE*	
Order	InputListBox	Prompt	Variable	(Empty)	Default(s)	(No action)	comment	True/False	True/False (exp or var)
3	*InputListBox*	*Select Analysis Area(s)*	*Ward*	*Select Ward from qry_Suburb*	*HOBSON*			*TRUE*	
Order	InputCheckBox	Prompt	List of Variables	(Empty)	Defaults	(No action)	comment	True/False	True/False (exp or var)
4	*InputCheckBox*	*Define Optimisation Operations*	*Organise Data,Transpose Data,Calc NPV,Optimise*		*0,0,0,1*			*TRUE*	

ID	Operation	ObjType /ObjectType	ObjName /ObjectName	Param /Parameter	FldName /Parameter2	Open	Comment	Included	InclAtRunT
Order	InputConfirmation	Variable	VariableName	Expression (definition)	Default	Open to define	comment	True/False	True/False (exp or var)
5	*InputConfirmation*	*Variable*	*DelOldData*	*IF(@FWP Section@=0, 'All data will be transposed, but you may want to prepare data for current year (@BaseYear@) only. Select Yes for current year only; No for all years; Cancel to stop the processing.',1)*				*TRUE*	

Input Calculation Variables

ID	Operation	ObjType /ObjectType	ObjName /ObjectName	Param /Parameter	FldName /Parameter2	Open	Comment	Included	InclAtRunT
Order	InputCalculation	Variable	VariableName	Expression (definition)	Default	Open to define	comment	True/False	True/False (exp or var)
6	*InputCalculation*	*Variable*	*Ins_Sect*	*Lookup('mSysSQL', 'Ins_Sect', 'SQLStatement')*			*global generic*	*TRUE*	

Output Calculation Variables

ID	Operation	ObjType /ObjectType	ObjName /ObjectName	Param /Parameter	FldName /Parameter2	Open	Comment	Included	InclAtRunT
Order	OutputCalculation	Variable	VariableName	Expression (definition)	Default	Open to define	comment	True/False	True/False (exp or var)
7	*OutputCalculation*	*Variable*	*Naasra_Pred_03*	*@A1@*([Naasra_Pred_02]) ^@A2@+@A3@*				*TRUE*	*@Surf_Age_03@=0 And @DoCalc@*

11.3 Use Runtime Variable

Runtime variables are directive definitions, so they are resolved on the fly. This determines their scope of usage.

Use Runtime Variables in Model
Model execution can be broken into a few steps:

- Initialising Model Operations (parse model definitions)
- Executing Model Defintions (parse variable definition and precompile models)
- Run other dedicated operations (such as optimisation etc)
- Runtime variables can only be used in the model initialisation stage.

Use Runtime Variables in Batch and Trigger
Since all operations in batch and trigger are simply calculation on the fly, runtime variables can be used throughout the whole object definitions.

Format to Use Runtime Variables
Runtime variables are simply defined with a variable name (a string per standard naming convention). But to use it, variable name will need to be delimited with @, make it as @VarName@.

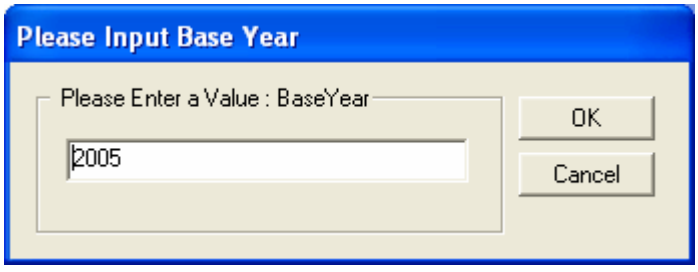
Here are some points to be careful of:

- Output variables can only be used inside triggers
- Input variables should normally used inside batches or models only
- Except model, normal variables and constant cannot be used within object definitions
- Expression definition for variables is the same for all objects no matter defined in-line within objects or as normal variables

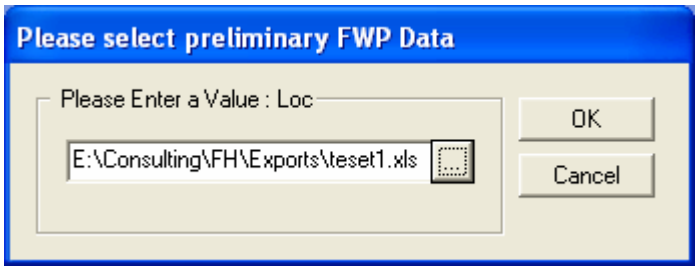
Since runtime variables can be used to build any strings or statement, necessary delimitation with quotes or type cast will need attention as well.

11.4 Interface for Getting User Input at Runtime

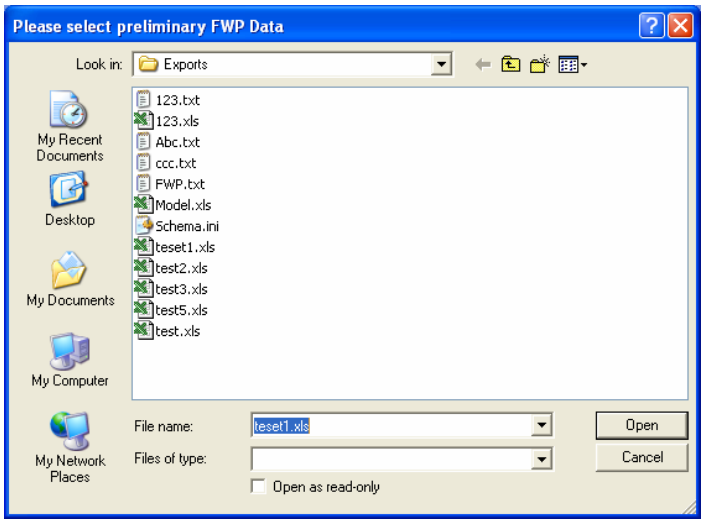
Input Text Box



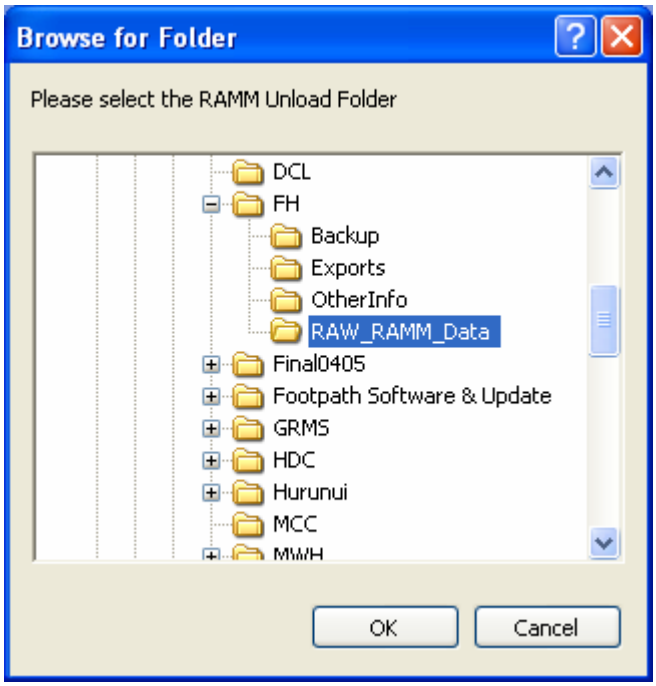
Input File Folder / Open / Save



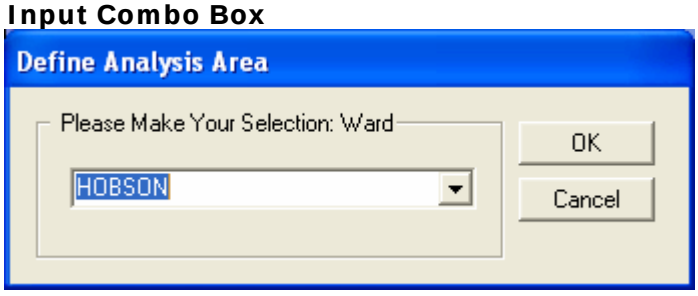
Input File Locator



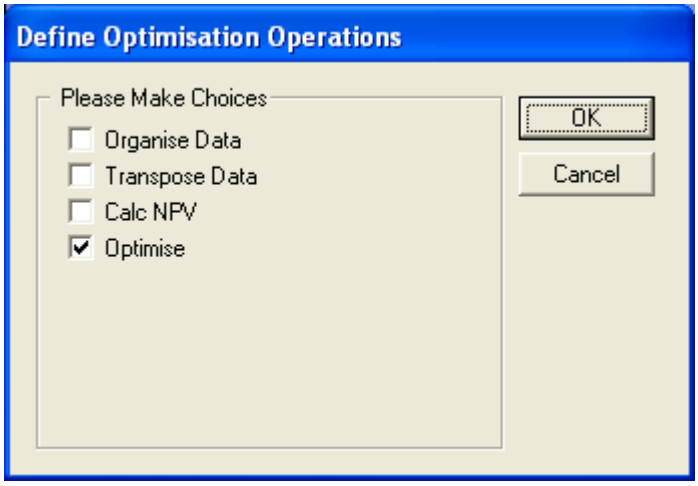
Input Folder Locator



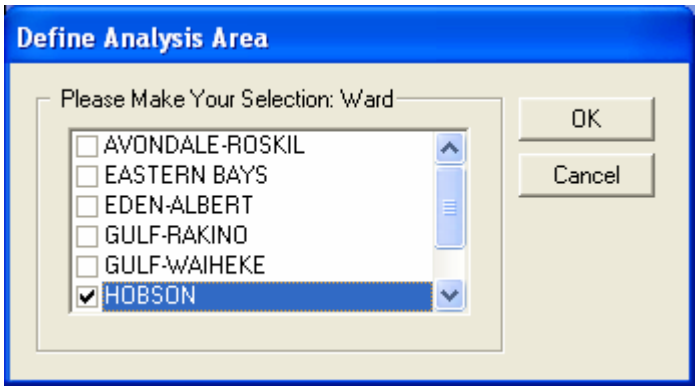
Input Combo Box



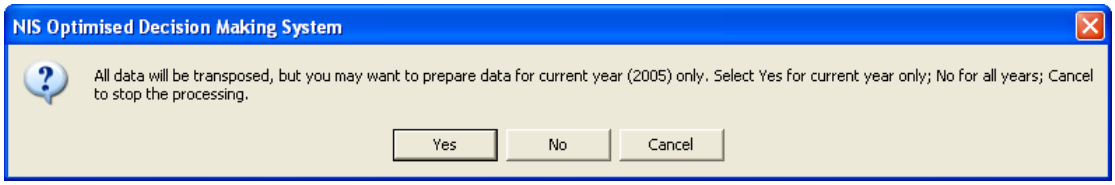
Input Check Box



Input List Box



Input Confirmation



11.5 Preset Runtime Variable Value

Input runtime variables, except InputConfimration variables, can be preset through Info Wizard and thus bypassed from user intervention during object execution. This is particularly useful for customisation of general model definition and some batch operation.

For detail, please refer to section *Working with Info Wizard.*

12 WORKING WITH MODELS

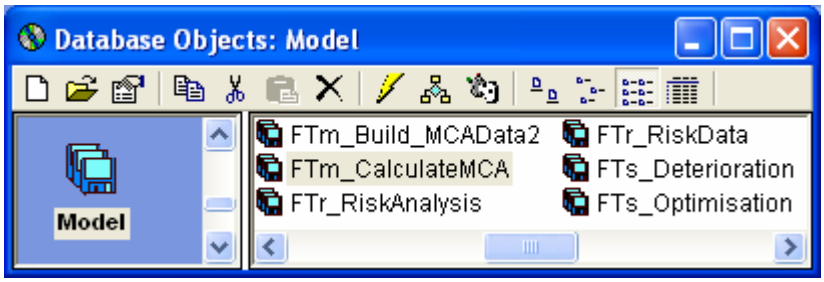
Models are used to carry out data calculation and processing. The possible use of Models could include:

- Predictive Modelling
- Optimisation
- Running templates
- Data calculation

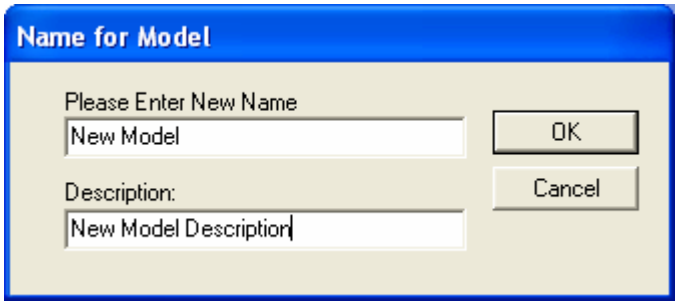
All available models are listed in the Database Objects: Model window which is opened by clicking Model in the left hand box of the Database Objects window

12.1 Create Model

To create a new model, select **Model** in the Left hand side box of Database Objects window and then click in the drop down menu of the Database Objects window.



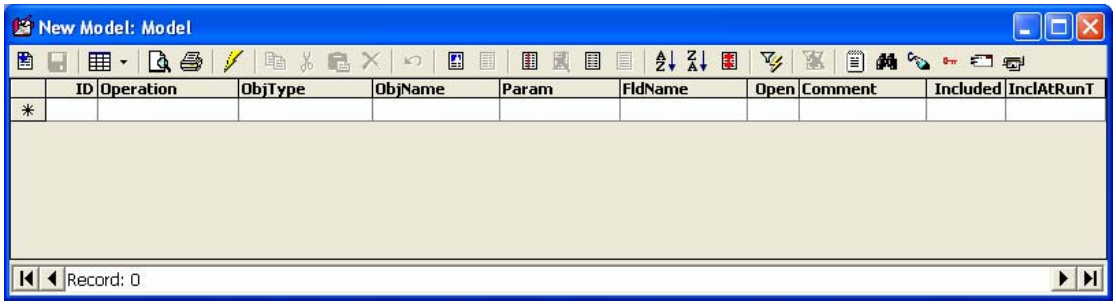
User should give unique model name when a window **Name of Model** appears.



If similar model is available user can create a new model by copying an existing model using and then . Later they can modify the models as required.

12.2 Define Model

A model definition window can be opened by selecting (highlighting) a model from the model list, and then clicking button on the top menu. It can be opened by simply double clicking the model name as well.



User needs to define various operations in the model table. The columns/fields included in the table include.

Table Caption	Operation Variables	Parameter / Values	Comments
ID	The operation identifier and in this sequential order (duplicates allowed).		
Operation	Allows selection of available operations in the operation library.		
ObjType	Allows selection of the following object type for operation:		
	Table		Allows to select a table
	Query		Allows to select a query
	Variable		Allows to select a variable
	Constant		Allows to select a constant
ObjName	The list of the predefined by **ObjType** objects will be available in the dropdown menu for selection		
Parameter	Allows to enter SQL statement as filter or for executing certain operation		
FldName	Allows to select a field from table or query defined by ObjName		
Open	Allows to open the given object defined by ObjName		
Comment	Any comments related to the operation		
Included	Whether or not to include the operation in the model run		
	0		Skip this operation
	-1		Include this operation
InclAtRunT	Whether or not to include the operation in run time		

List of Operations related to ODM process

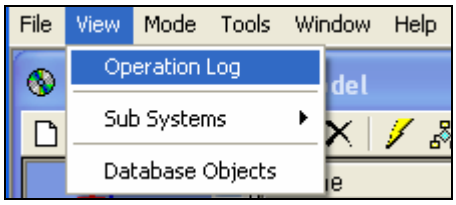
General Operations		
LookupData	Table/Query to be used for lookup	
Initialise	Variable	Calculate the expression defined as **Initiation** expression of a given variable
Calculate	Variable	Calculate the expression defined as **Calculation** expression of a given variable
Loop function		
LoopStart	Table/Query Or LoopVar	Defines parameter used by the loop
LoopEnd		End of the loop
IfStart	Boolean Variable	Boolean variable. Block if start
ElseStart		Block else start
IfEnd		Block if end
SkipOn	Boolean Variable	
BreakOn	Boolean Variable	
FilterOn	Boolean Variable	
Specific Operations related to Predictive modelling and alternatives generation		
InputData	Table/Query	Defines the name of source data table or query
OutputData	Table	Defines the name of table where the output data to be written
MaximumAlternatives	MaxNum	Maximum number of strategies
AllowDoNothing	0/-1	Do nothing strategy
ClearReset		Clear resetting for next calculation

LoadReset		Load reset for strategy generation
NewUpdate		Add a record and Save
Update		Save the current record
WriteReset		Save the generated stragety
Reset		Do reset operation
Specific Operation Related to Prioritisation and Optimisation Functions		
PriDoPrioritisation		Run Prioritisation Routine
OptOrganiseData		To run data organisation process as part of Optimisation process
OptTransposeData		To run data organisation process as part of Optimisation process
OptCalculateIBC		To Calculate IBC
OptDoOptimisation		To undertake final optimisation

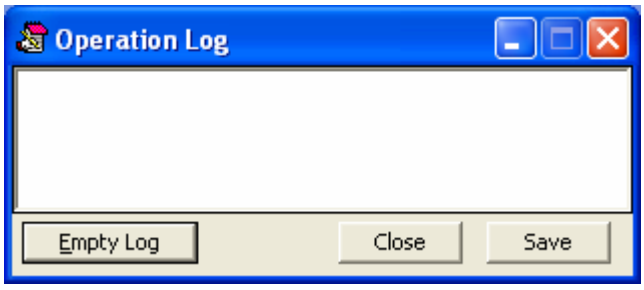
For more model related statements, please see section ***System Built-in Statements***

12.3 Test Model

For testing of the model user can open the Operation Log window by selecting **View / Operation Log.**



Example of a blank operational log window is given below. User can empty the messages from the previous operations by clicking on **Empty Log** window. Similarly the messages can be saved to an external file for future references using **Save** button.



To execute the model, click on  button on the top menu. The operations executed and any error message will be logged and displayed in the operation log window. For debugging purpose, the user can change the statement to ShowMessage, then values or expression can be displayed in operation log.

12.4 Use Model

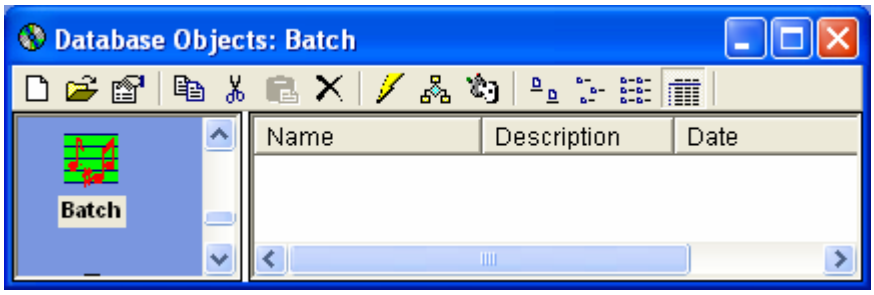
Models are generally included in **Trigger, Batch or Analysis** objects or in sub system navigator to be used for various procedures.

13 WORKING WITH BATCHES

Batch object in NODEM is used to run a number of operations as a batch, normally without too many user interventions. Runtime user input may required inside batch as well to increase the flexibility of batches.

13.1 Create Batch

To create a new batch, select **Batch** in the Left hand side box of Database Objects window and then click in the drop down menu of the Database Objects window.



User should give a unique name for the Batch when a window **Name of Model** appears.

13.2 Define Batch

A batch definition window can be opened by selecting (highlighting) a Batch available in Batch list window, and then clicking button on the top menu. It can be opened by simply double clicking the Batch name as well.

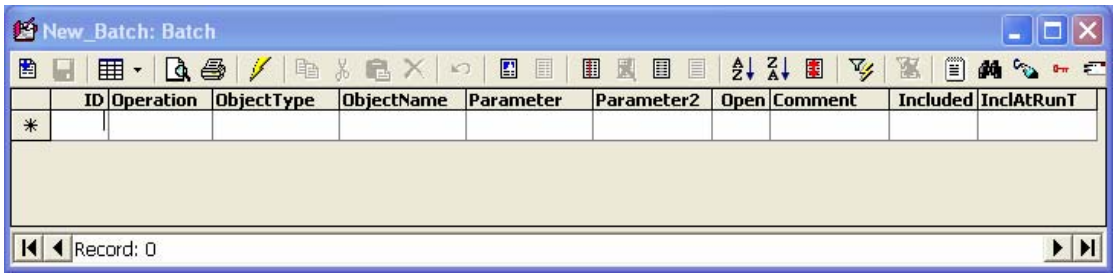
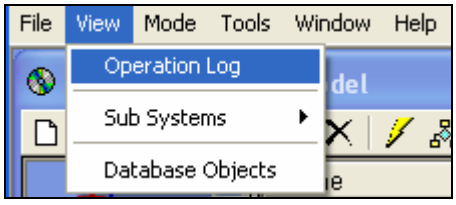


Table	Operation	Parameter /	Comments

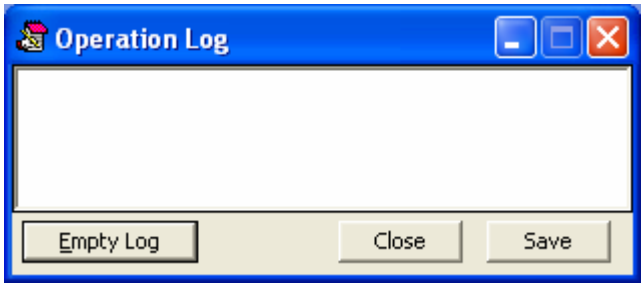
Caption	Variables	Values	
ID	The operation identifier defined by an Integer (Duplicate allowed) . The operations are executed in ascending sequential order based on number defined in this column		
Operation	Allows selection of the available operations in the operation library.		
	Open		
	Run		
	Delete Object		Delete an Object
	Delete Data		Delete data in a table or query
	Join		Join Table or Query
	Validate		Run Validation object
	Export		Export data
ObjType	Allows selection of the following object type for operation:		
	Table		Allows to select a table
	Query		Allows to select a query
	Form		Allows to select a form
	Report		Allows to select a report
	Document		Allows to select a document
	Sectioning		Allows to select a Sectioning
	Transformation		Allows to select a Transformation
	Batch		Allows to select a Batch
	Model		Allows to select a Model
ObjName	The list of the predefined by **ObjType** objects will be available in the dropdown menu for selection		
Parameter	First parameter for an operation. Normally a SQL statement as filter or for executing certain operation		
Parameter2	Second parameter for an operation. Normally default value, file name etc		
Open	Allows to open the given object defined by ObjName		
Comment	Any comments related to the operation		
Included	Whether or not to include the operation in the model run		
	0		Skip this operation
	-1		Include this operation
InclAtRunT	Whether or not to include the operation in run time		

13.3 Test Batch

For testing of the model user can open the Operation Log window by selecting **View / Operation Log.**



Example of a blank operational log window is given below. User can empty the messages from the previous operations by clicking on **Empty Log** window. Similarly the messages can be saved to an external file for future references using **Save** button.



To execute the Batch, click on button on the top menu. The operations executed and any error message will be logged and displayed in the operation log window.

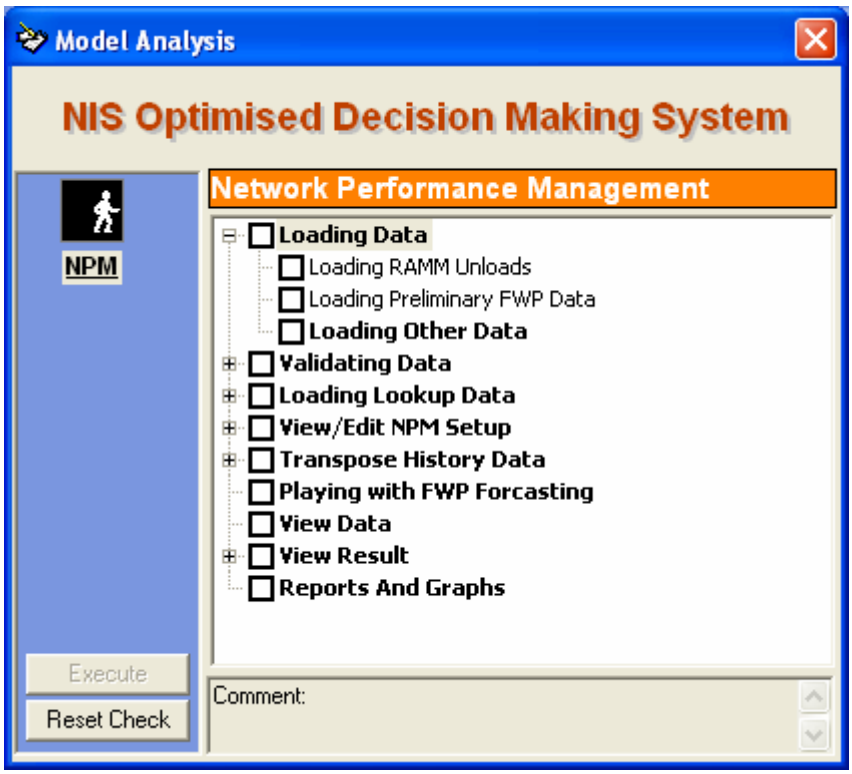
13.4 Use Batch

Batches are generally included in **Analysis** objects to be used for various procedures.

14 WORKING WITH ANALYSES

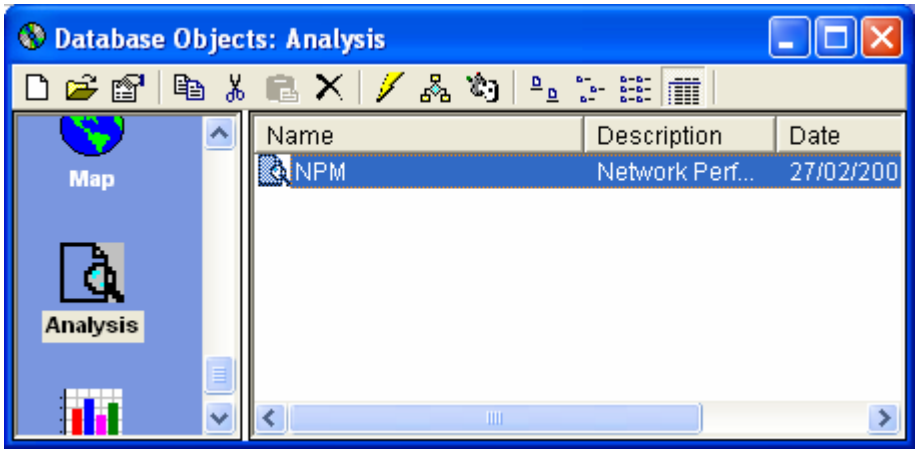
Analysis object in NODEM is used to organise complex modelling and analysis in logic order and execute in a step by step base. Any type of object can be included, from data object (for view or data entry), through data modelling and analysis to output presentation (reports graphs and maps).

A typical analysis at run time is as following.

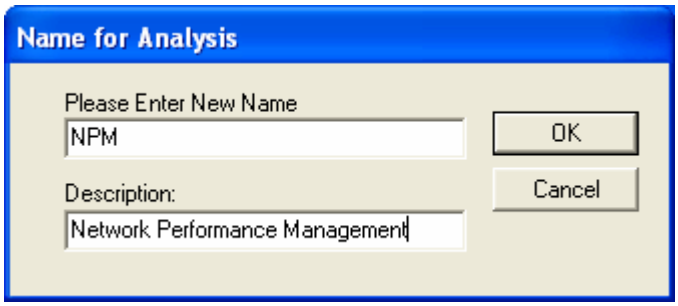


14.1 Create Analysis

To create a new batch, select **Analysis** in the Left hand side box of Database Objects window and then click in the drop down menu of the Database Objects window.



User should give a unique name for the Analysis when a window **Name for Analysis** appears.



14.2 Define Analysis

An analysis definition window can be opened by selecting (highlighting) an Analysi available in Analysis list window, and then clicking button on the top menu. It can be opened by simply double clicking the Analysi name as well.

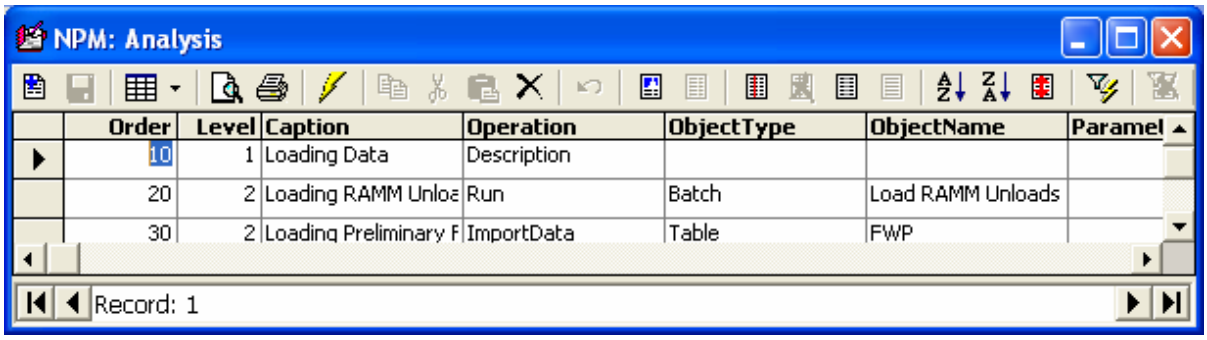
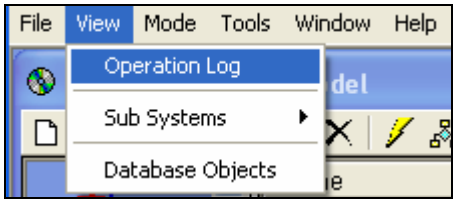


Table Caption	Operation Variables	Parameter / Values	Comments
Order	The sequential order for building analysis interface		
Level	The indentation level when display		
Caption	The caption for an operation on dispaly		
Operation	Allows selection of the available operations in the operation library.		
	Description		Used for display a group of operations
	Open		Open data object
	Run		Run operational object
	Delete Object		Delete an Object
	Delete Data		Delete data in a table or query
	Join		Join Table or Query
	Validate		Run Validation object
	Export		Export data
	ImportData		Import Data to a table
ObjType	Allows selection of the following object type for operation:		
	Table		Allows to select a table
	Query		Allows to select a query
	Form		Allows to select a form
	Report		Allows to select a report
	Document		Allows to select a document
	Sectioning		Allows to select a Sectioning
	Transformation		Allows to select a Transformation
	Batch		Allows to select a Batch
	Model		Allows to select a Model

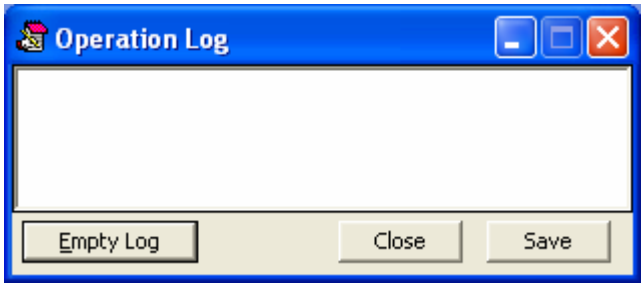
	Chart		Allows to select a chart
ObjName	The list of the predefined by **ObjType** objects will be available in the dropdown menu for selection		
Parameter	First parameter for an operation. Allows to enter SQL statement as filter or for executing certain operation		
Parameter2	Second parameter for an operation. Normally default value, file name etc		
Open	Allows to open the given object defined by ObjName		
Comment	Any comments related to the operation		
Included	Whether or not to include the operation in the model run		
	0		Skip this operation
	-1		Include this operation
InclAtRunT	Whether or not to include the operation at run time		

14.3 Test Analysis

For testing of the Analysis user can open the Operation Log window by selecting **View / Operation Log.**



Example of a blank operational log window is given below. User can empty the messages from the previous operations by clicking on **Empty Log** window. Similarly the messages can be saved to an external file for future references using **Save** button.



To execute the Batch, click on  button on the top menu. The operations executed and any error message will be logged and displayed in the operation log window.

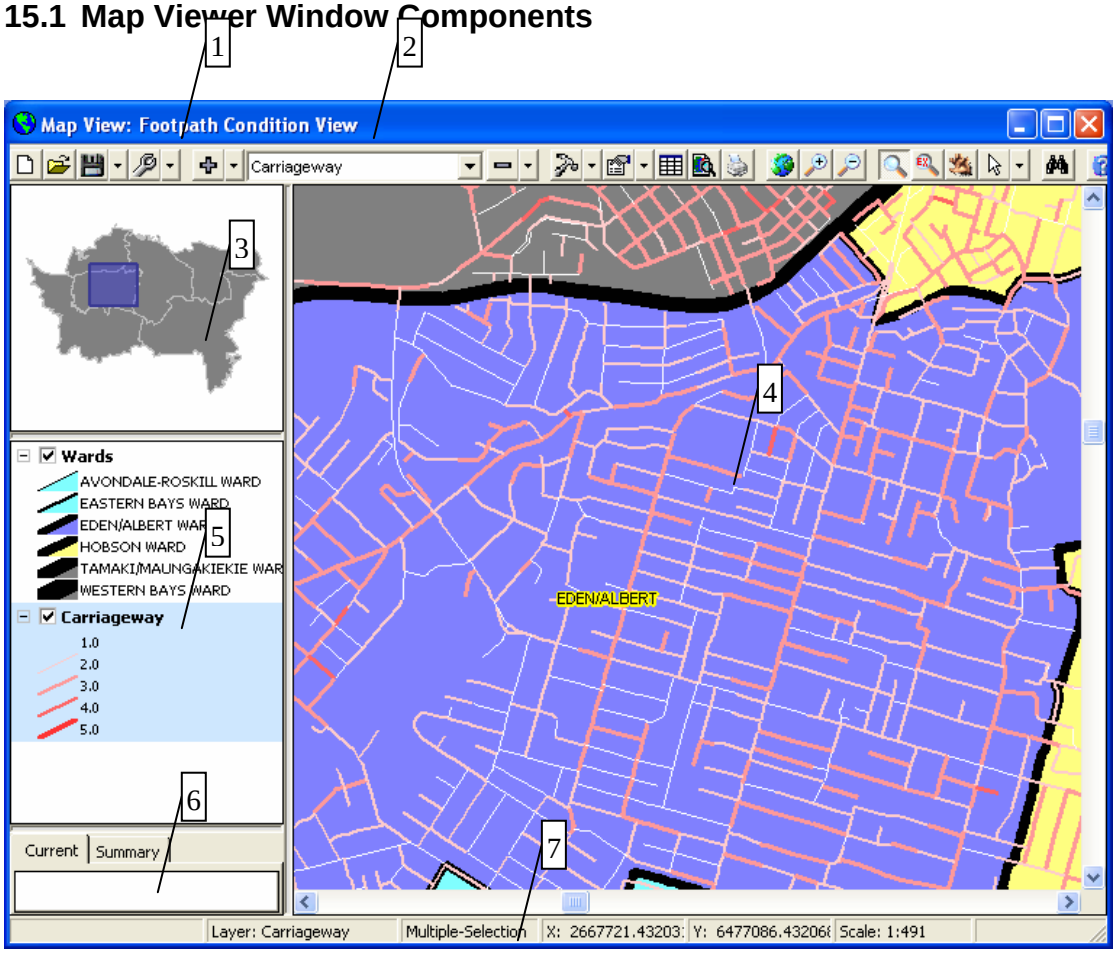
14.4 Use Analysis

Analysis objects can be executed from database object browser, but mainly run from sub system navigator.

15 WORKING WITH THEMATIC MAPS

NODEM comes with a built-in map viewer which can support most of the industrial standard GIS data format. It can directly support ArcView project file, MapInfo project or workspace file etc. For map layers, it can support most of the standard field format including Shp, Tab, Cad, Jpeg, Tiff etc. And most importantly, NODEM map viewer can make Geo database and directly use data inside the database to dynamically generate and present timely information in GIS format. NODEM can also export GIS data to feeder other GIS system.

15.1 Map Viewer Window Components



Following are all components for a typical set of map layers

1. Map tool bars, provides all functions related to map operations
2. List of layers on map. Select an item from the list will navigate to the right layer
3. Mini map, when mini map has map layer, it is used to navigate to desired location when the main map window is zoomed in
4. Main map window, all map layers are drawn here
5. Map layer legend
6. Map information for selected features for a given map layer
7. Status bar with map layer information

Following tables describe the functionality in NODEM map viewer.

Map Tool Bars	provides all functions related to map operations
New	Create new project (or configuration file)
Open	Open an existing project file
Save	Project (configuration) Save (default) /Save operation
Database Tools	Tool for Load SQL Layer (saved database layer, default) and Save as SQL layer (save external map layer)
Add Layer	Add External Layer (external files), Add SQL Layer (default), and Add Layer (current layer to Mini map)
Carriageway	Map layer selection combo box
Delete Layer	Delete Layer (delete layer from map display only, default), Delete Layer from Mini Map display
Map Tools	Layer Property (current layer, default), Calculate Extent (calculate map extent), Export Layer to File (save layer to outside for other GIS system)
Map Properties	Map View Properties (default), Hide/Show Mini map, Hide/Show Legend, Hide/Show Info (current selection)
View Data	View map data (if selection exist, then only selected data). Support SQL layer only (not external data file)
Preview/Print	Preview/Print map
Full Extent	Display full map
Zoom	Map zooming modes
Pan	Map moving (can also use slider bar in main map view)
Select Mode	Deselect All (deselect in current layer), Multi/Single Select, Select by Point/Line/Polygon/Circle/Rectangle
Find Shape	Find/filter map shapes satisfying a criteria
Mini Map	For area navigation. The main map display location and rectangle in minim map location are synchronised. So move the rectangle in the mini map will relocate the main map display location, and zoom or pan the main map will relocate the rectangle in the mini map as well
Legend	Map layer and property displayed as legend. Double click will invoke the definition of layer properties; hold and move a layer will change the order of the layer in map; highlight a layer will make it current layer (same as the Map Layer Selection combo box)
Map Info	Current (display the data for current shape), or Summary (for all selected shapes)

15.2 Prepare GIS Data

Currently NODEM comes only with Map Viewer for thematic map presentation. All GIS shapes thus will need to be prepared through GIS software or GPS data collection. Fortunately this is not a problem at all as a lot of GIS shape data already exist in standard format of one way or the other.

NODEM map viewer can work with most of the standard GIS file format directly or convert the data and save into database. Upon save, the map layer will be registered inside NODEM, and two data tables will be created with a name of the layer plus a suffix "_FEA" (for feature) or "_GEO" (for geography), and these tables are linked through a unique identification field (UID).

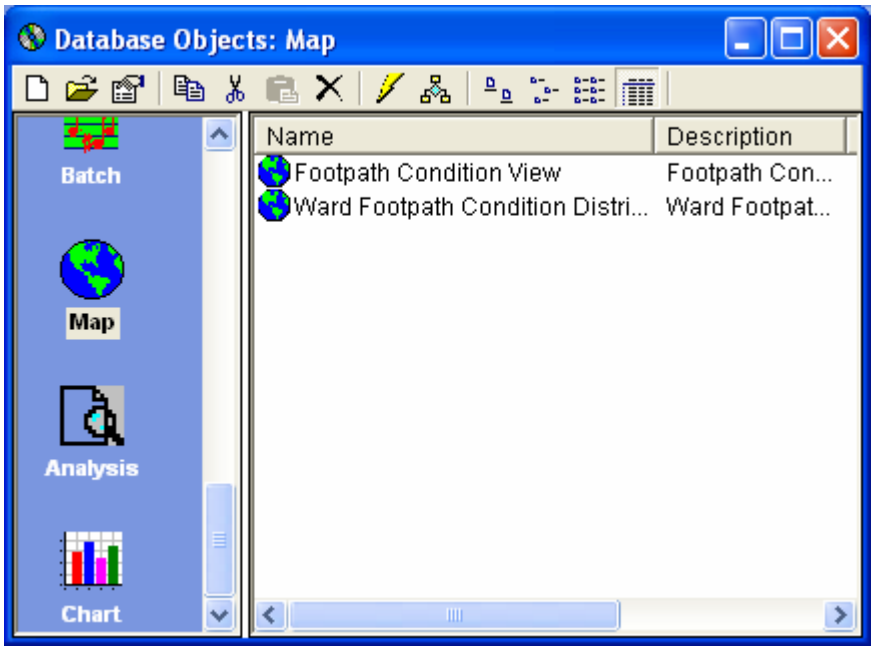
Data inside the database thus can be very easily linked to the _GEO table through newly generated table or query. As long as a table or query named as a layer plus suffix "_FEA"

with linkage (UID), NODEM Map Viewer will treat it as the layer feature definition, and will be loaded to the Map Viewer.This makes the NODEM Map Viewer extremely flexible to work with different types of data. For detail, please see section *Working with Tables* and *Working with Queries.*

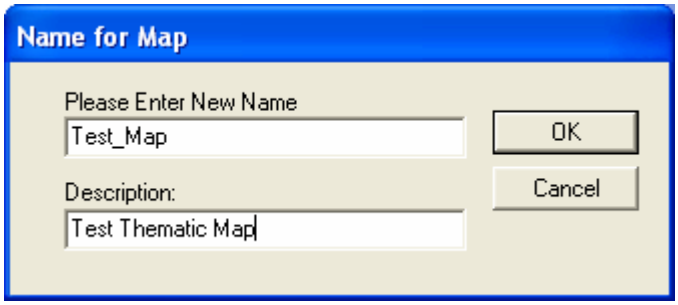
15.3 Create Thematic Map

Once all necessary GIS layers are defined (and saved inside the database), the user can define map of different themes to satisfy various purposes. Thematic map can be defined by user of analyst level or above. Thematic map is actually a configuration of map layers inside the sub systems

Open database object browser and locate the Map icon in left list view. All existing thematic maps will be displayed.



Select to create new thematic map. It will be added into the list.



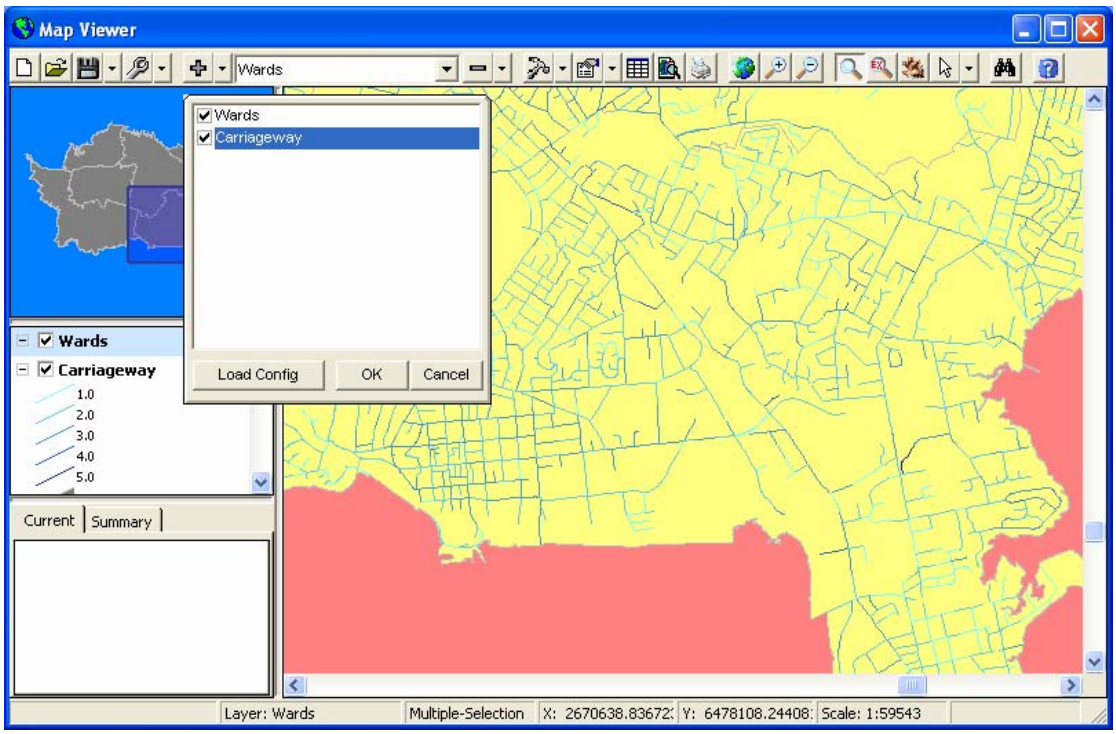
15.4 Define Thematic Map

Definition for a newly created thematic map is empty and need to be made through the NODEM Map Viewer. Defined thematic map can be modified as well.

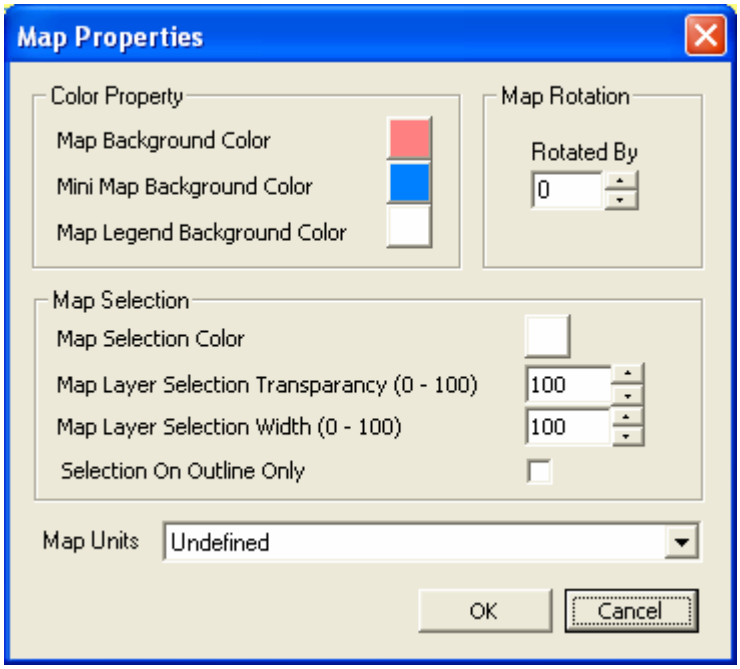
Select  to open a thematic map. If configuration existing for a thematic map, it will be used for loading map layers. It not yet, then the user will need to select the layers and define the properties for each layer.

The user can define, modify and at last save the configuration (as an external file). Once it is done, the configuration can be used for the thematic map.

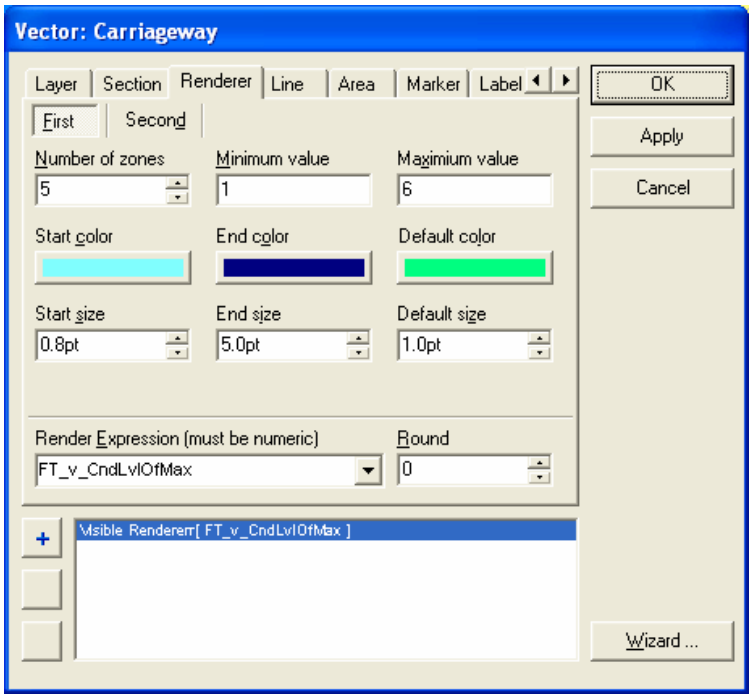
Load Map Layer and Configure Thematic Map



Map View Property



Define Map Layer Properties

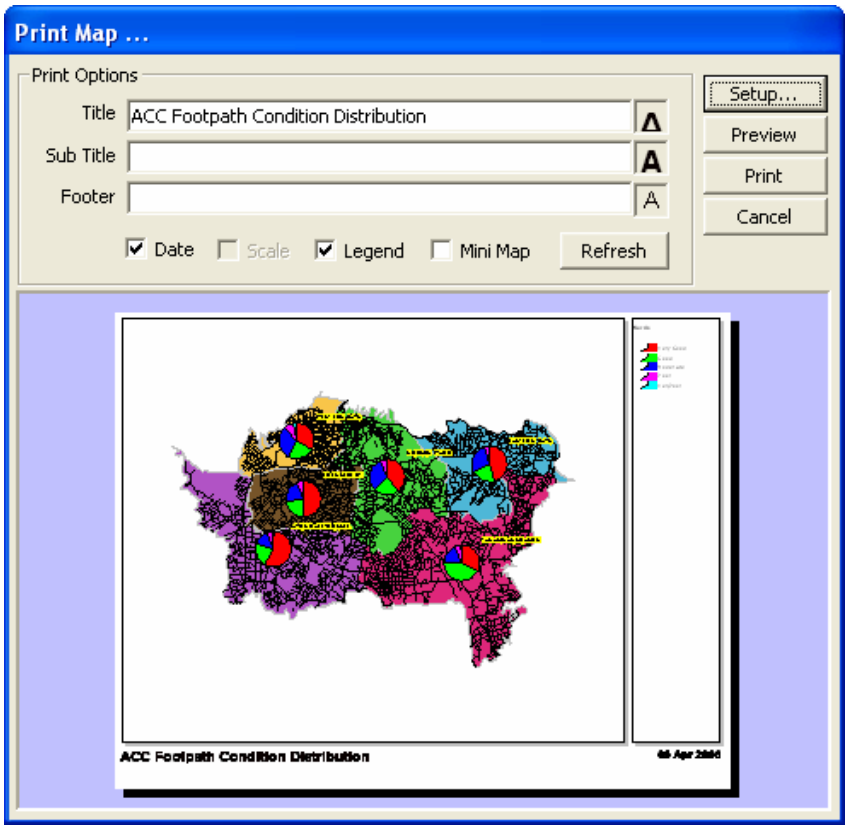


15.5 Use Thematic Map

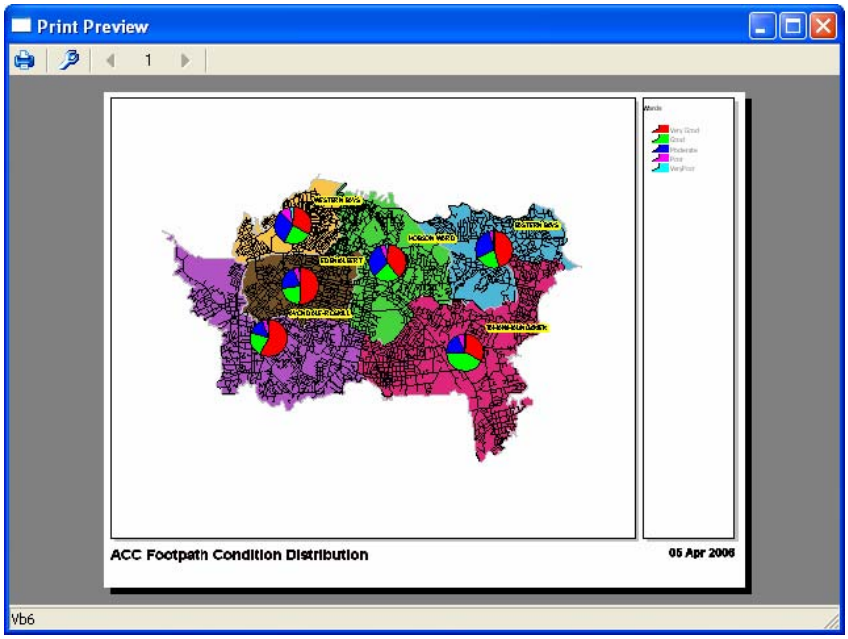
Thematic maps are mainly used in analysis object and sub system navigator. Although the configuration (saved externally) can be used directly from the Map Viewer for display.

Printing Thematic Map

Click to print preview the map. Before printing, the user can do some basic printing setup and preview the map.



Click **Preivew** for preview, **Print** to print out the map.

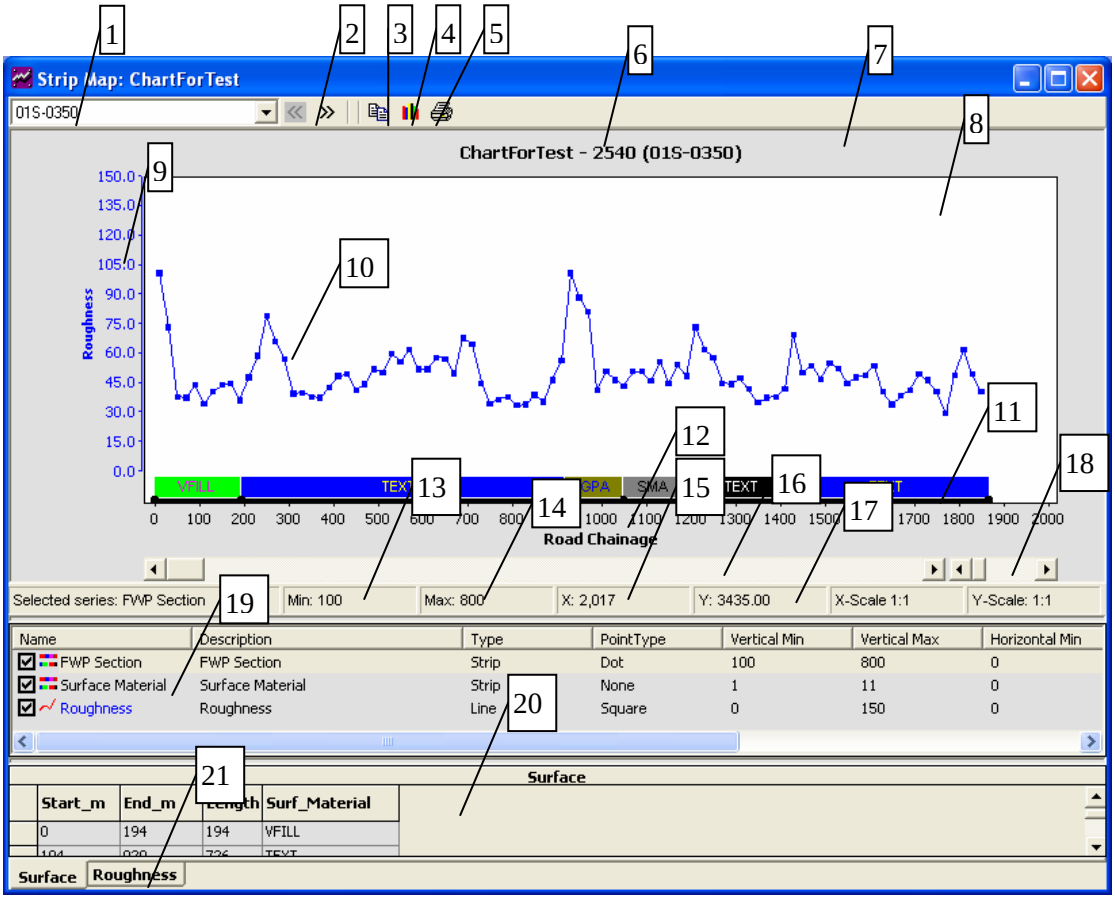


16 WORKING WITH STORED CHARTS

Stored charts are graphic view of data of different features with certain relations. They will be configured to run, and limited data operations are allowed depending on the way a chart is configured.

16.1 Chart Window Components

Following are all components for a typical chart.



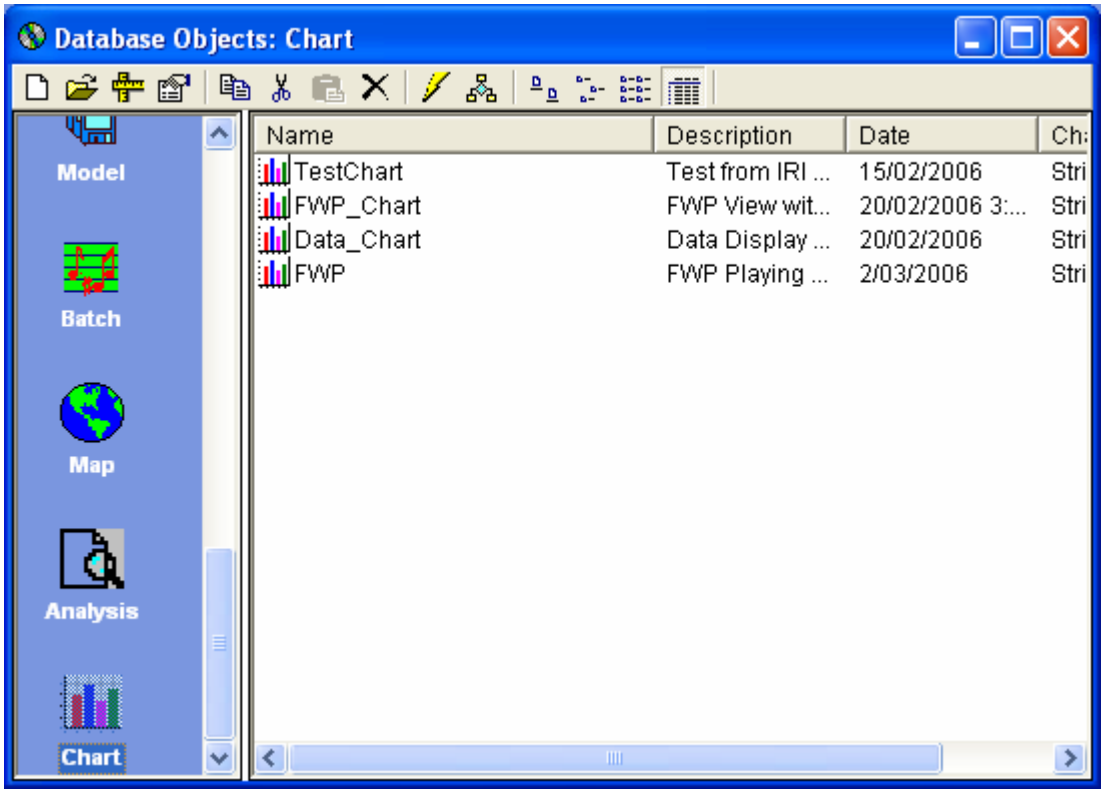
The components show above are:

1. *Chart Group Navigator*: a list of group data item IDs or Names (such as Road_ID/Road_Name). To switch between them, set focus to the combo box, then press CTLR+V. Selecting a item from the list will navigate the to the chart for that item
2. *Navigation Buttons*: including move to previous (<<) and move to next (>>0).
3. *Copy*: copy chart to clipboard. It will only copy the chart area
4. *Save Char to File*: save chart area to a bmp file
5. *Print*: print chart area to file
6. *Chart Title*: an combination of user define chart caption plus group item name
7. *Chart Area*: the area chart and axes are drawn, not including sliding and scaling bars
8. *Chart Drawn Area*: area where all chart series are drawn
9. *Y Axis*: y axis and title. A chart may have a lot of Y Axes
10. *Chart Series*: a line chart series with square shape marking data points
11. *Chart Series*: a strip series
12. *X Axis*: chart X axis. Defined for all chart series
13. *Min Value*: minimum value for current series
14. *Max Value*: maximum value for current series
15. *X*: x value per X Axis
16. *X Slider*: to move the chart in horizontal direction

17. *Y*: y value per chart scale. Not a meaningful value for series
18. *X Scale*: to change the scale value of x axis
19. *Legend*: chart series legends, including a lot of different attributes for a chart series definition. A Chart series can be turned on or off from display
20. *Chart Data*: chart data display in data grid format
21. *Chart Data Group Tab: series can be grouped into different data group. Data from different data source cannot be put into same group.*

16.2 Create Stored Chart

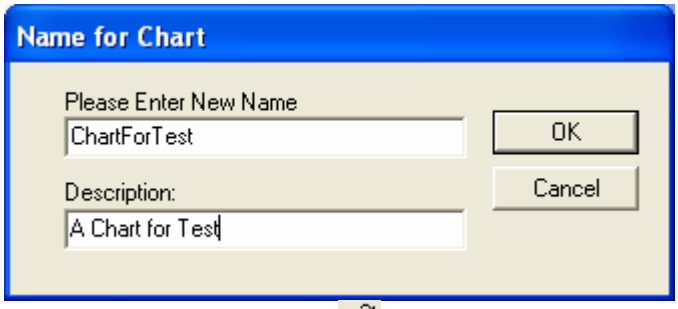
Login as NODEM configurer and select Data Administration mode, then start database object browser and locate the Chart icon in left list view. Then all existing charts will be displayed.



Here the user can:

Create New Chart

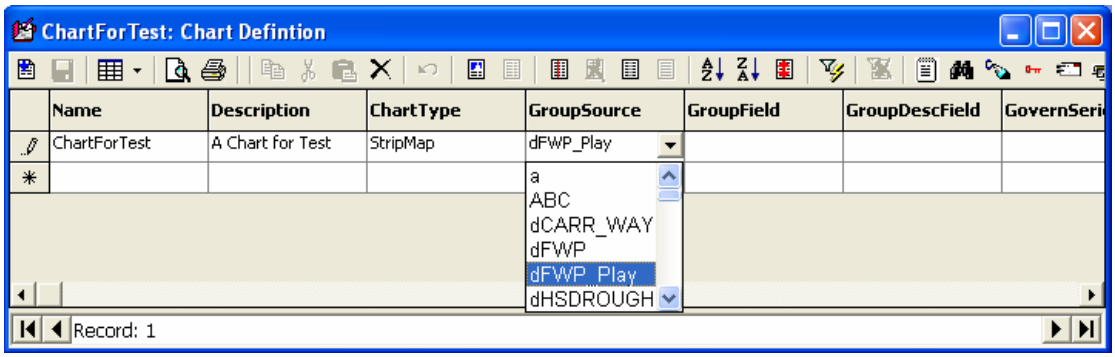
Define a new chart with name and description. The newly defined charts will be added to the list for design.



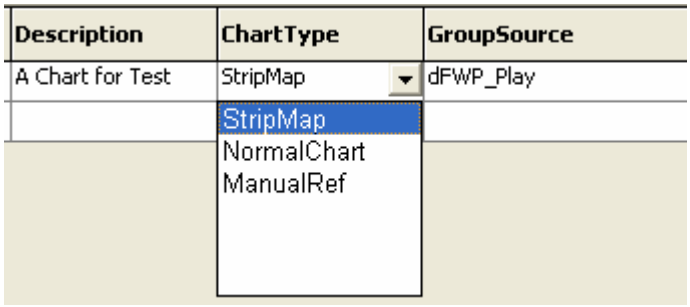
Open Chart Definition

Define chart head information. This is a single record definition with a lot of fields to define. A selection list or button click operation has been supplied whenever applicable,

including value selection, list of data sources and fields, colour selection, filter definition etc.

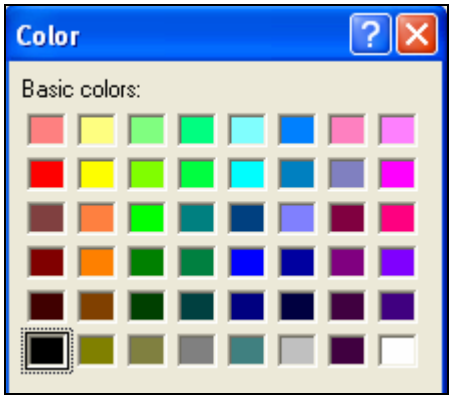


List of Value as Selection List

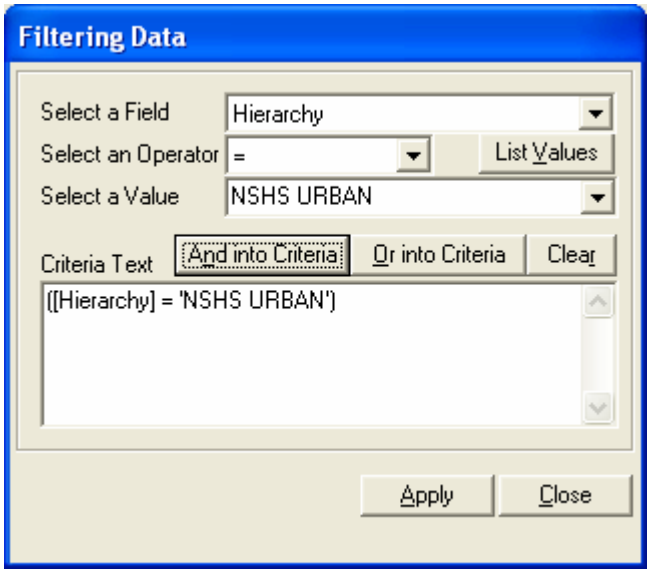


List of Tables as Selection List

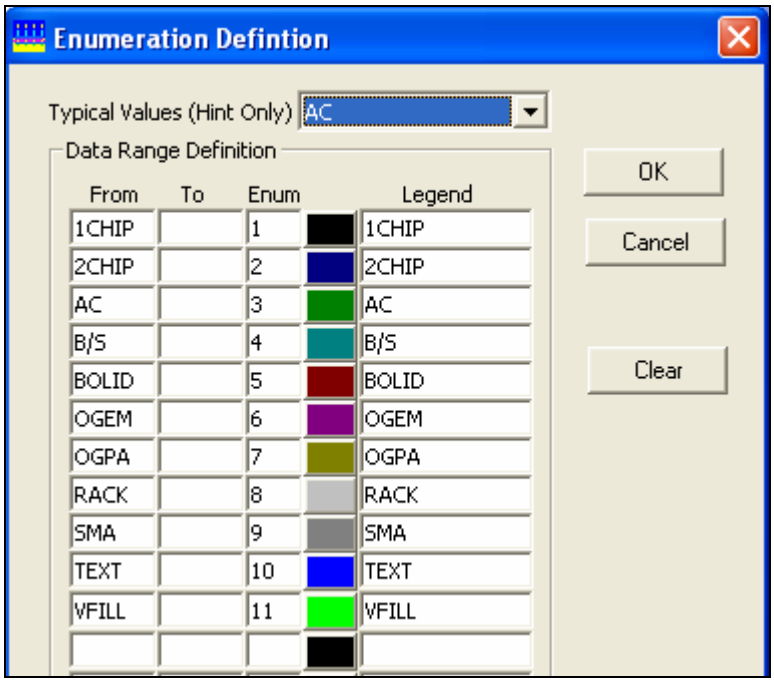
Colour Selection



Filter Definition



Enumeration Definition



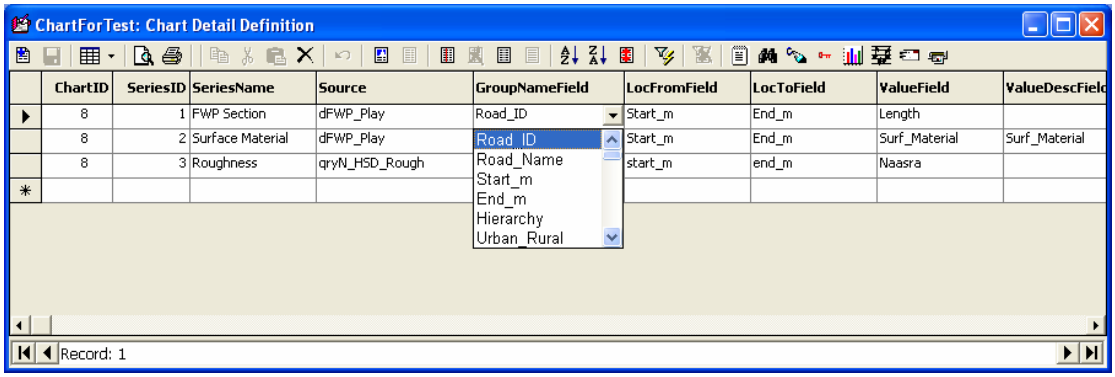
Following are description about all the field definitions:

- **Name**: Chart name to be identified as object identifier. Must be unique through all charts
- **Description**: Chart description for information
- **ChartType**: type of chart. Please select *StripMap*, as this type support all functions and can be flexibly configured to various view styles.
- **GroupSource**: chart group list source table or query name, used to make as list of items (such as Roads) for chart navigation.
- **GruopField**: chart unique group item field name (such as Road_ID)
- **GroupDescField**: chart group item description field name (such as Road_Name). To be useful, GroupDescField need to be unique as well
- **GovernSeriesName**: the main series to govern the X-Axis for chart view, chart horizontal sliding and scaling. This must be the series name inside the chart detail definition.

- **XAxisTitle**: x axis title
- **InputData**: reserved
- **OutputData**: reserved
- **AllowRefine**: reserved
- **ForeColor**: chart foreground colour
- **BackColor**: chart background colour
- **MinMarginPct**: minimum horizontal margin for chart area (including both chart and Y Axis) to left and right chart frame area.
- **FilterString**: a string to filter the selection of group item list
- **OptimiseDataSource**: join series from same data source together or not. An internal flag, and can left as -1 (True) all the time.
- **Left**: left position of chart window when open
- **Top**: top position of chart window when open
- **Height**: height of chart window when open
- **Width**: width of chart window when open
- **HSplit**: horinzontal split bar position (top)
- **VSplist**: vertical split bar position (left)
- **Tool**: toolbar to be customised
- **Design**: toolbar to be customised
- **Property**: toolbar to be customised (display or not)
- **View**: toolbar to be customised
- **Save**: toolbar to be customised
- **Save**: toolbar to be customised
- **Undo**: toolbar to be customised (display or not)
- **Date**: the date record defined

16.3 Define Stored Chart Detail

Design Chart  : define chart series detail. A chart must have at least the governing series and other series for display. Each series is a single record definition with a lot of fields to define. A selection list or button click operation has been supplied whenever applicable, including value selection, list of data sources and fields, colour selection, filter definition and enumeration definition etc.



Following are description about all the field definitions:

- **ChartID**: Chart ID from chart (header) definition
- **SeriesID**: Series ID of a series in a chart definition
- **SeriesName**: name of a series. One of the names must match the *GovernSeriesName* inside chart definition
- **Source**: series data source.
- **GroupNameField**: the field with value matches the *GroupNameField* definition inside chart definition. A must have
- **LocFromField**: location from field (such as Start_m) in x axis. A must have

- **LocToField**: location to field (such as End_m) in x axis. A series may not have LocToField.
- **ValueField**: the Y axis value, a must have
- **ValueDescField**: description of Y value. May be the same as ValueField.
- **EnumDefinition**: use enumerated value to represent series ValueField. Enumeration is typically used for string or date value display, but can also be used for number as well. In latter case, either discrete value (if series value is discrete) or range (discrete or continuous value) can be used. Up to 16 enumerated values can be used with represented value or colour
- **SeriesType**: type of the series, selected from a list of values, including point, line, bar, strip and step
- **UseEnumValue**: use enumerated value to draw or not if existing
- **UserEnumColor**: use enumerated colour definition to draw or not
- **ForeColor**: chart series fore colour
- **BackColor**: chart series background colour.
- **DrawWidth**: chart series drawing width
- **DrawStyle**: chart series drawing style, selected from a list of values including *SolidDash, DashDot, DashDotDot, Dot, InsideSolid, Invisible*
- **FillStyle**: chart series drawing filling style for series of bar and strip types, selected from a list of values, including Solid, Cross, Diagonal Cross, DownwardDiagonal, UpwardDiagonal, Transparent, HorizontalLine, VerticalLine
- **SeriesPointType**: point type of chart series, selected from a list of values, including *None, Dot, Square, Plus, Cross, HollowDot, HollowSquare*
- **ShowValueDesc**: display value description or not (depending on ValueDescField as well)
- **BandLocPct**: chart series drawing bottom position in terms of percentage of height of chart drawing area
- **BandHeightPct**: chart series drawing height in terms of percentage of height of chart drawing area.
- **BandID**: band id or the series. All chart series in the same band will be drawn together and share some common attributes, such as . Same band (ID) should normally have same BandLocPct and BandHeightPct, but it may differ in specific occasion
- **BandBottomLined**: draw a bottom line for a band, normally at BandLocPct. It can be drawn at the SplitLevel when a series is with split for Y value
- **ShowSereislegend**: display series legend or not, selected from a list of values, including not show, show title only, show axis (with title)
- **SereisTitle**: title of series for axis
- **YScaleMode**: scale mode of series values, selected from a list of values, including *NoScale* – automatically pick up the min and max value from the data source without any rounding, *AutoScale* – automatically pick up the min and max value for a particular group item (eg a road) with rounding, *ManualScale* – use manually set scale limits, *RangeScale* – automatically pick up the min and max value with rounding from the data source (thus for all group items), *Dynamic* – use the min and max value for a particular group item without rounding
- **YMinValue**: series manual set lower limit
- **YMaxValue**: series manual set upper limit
- **WithSplit**: series with a split for value above and below SplitLevel
- **SplitLevel**: a threshold value for a chart series. It will better reflect the data for series of bar type, all other types are anyway continuous drawing
- **DataGroupName**: for series data to be selected in tab format. Different series can be put in the same group if they are from the same data source
- **AllowEditRef**: allow to modify the Governing series definition of x axis
- **AllowEditData**: allow to modify series data
- **EditMode**: mode for edit, selected from a list, including 0 – ReadOnly, 1 – Allow Edit, 2 – Allow Add, 3 – Allow Delete. EditMode will dependent on AllowEditData as well

- **FullSource**: include whole source table for the series or only the series related fields
- **Display**: display series or not. If not displayed, it can be turned on for display at runtime. The governing series should be displayed all the time
- **ExtraData**: extra data field(s) to be included in the data group for reference.
- **ColPopup**: lookup definition for series field values
- **ColWidth**: column width when displayed in data grid format
- **Format**: series value display format in data grid
- **FilterString**: a filter string to filter series data to be included
- **Date**: time when the series to be created
- **Included: a series may have been defined but not included at runtime. The governing series should not be excluded.**

16.4 Use Stored Chart

Stored chart can be run directly from database object browser, or from controlled operations, such as navigator, batch, analysis etc.

Run from Object Browser

Select  to run the highlighted chart. All group items (such as Roads) will be included for the user to view chart against.

Call to Run from Other Object

Stored chart can be configured to run inside navigator, even with filter through list item selection from navigator. The configuration format can be:

Order	Tab	Item	Value	Value1	Other
825	NPM	Chart	FWP	RunOrRunFilter@[Road_ID]	FWP Play with Graph

For detail, please see *Working with Sub System Configuration*

Stored chart can also be called from analysis and batch, the format is basically as:

Operation	ObjectType	ObjectName
Run	Chart	FWP

17 WORKING WITH TRIGGERS

Not like database trigger coming with database management system, NODEM trigger is used to automatically invoke other operations when data changed. Current two types of triggers are supported:

- **OnUpdateField**: happens after column updated but before record updated
- **OnUpdateRecord**: happens after record updated

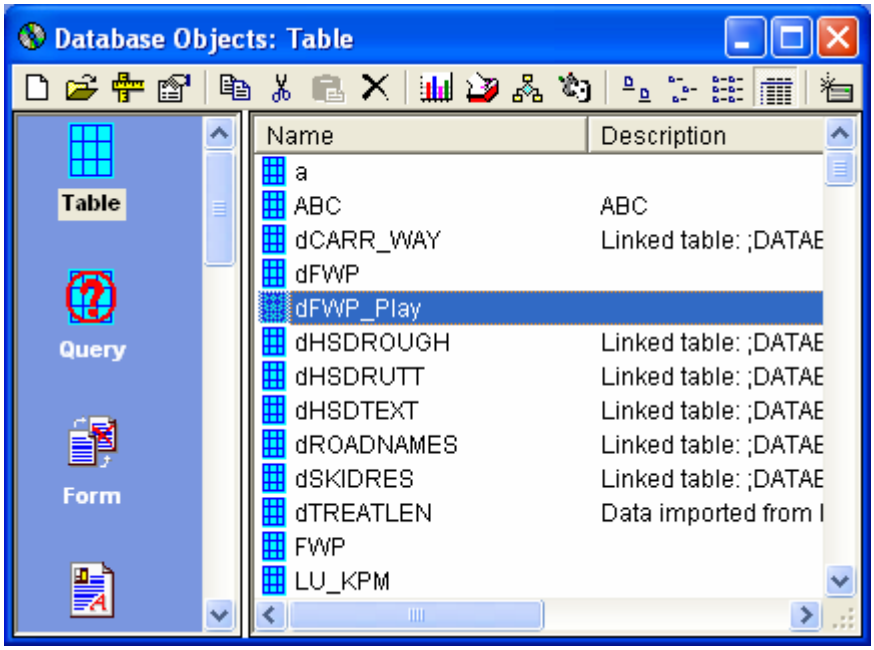
NODEM trigger can be used to do calculation using runtime variables, can call batch, model and other operations.

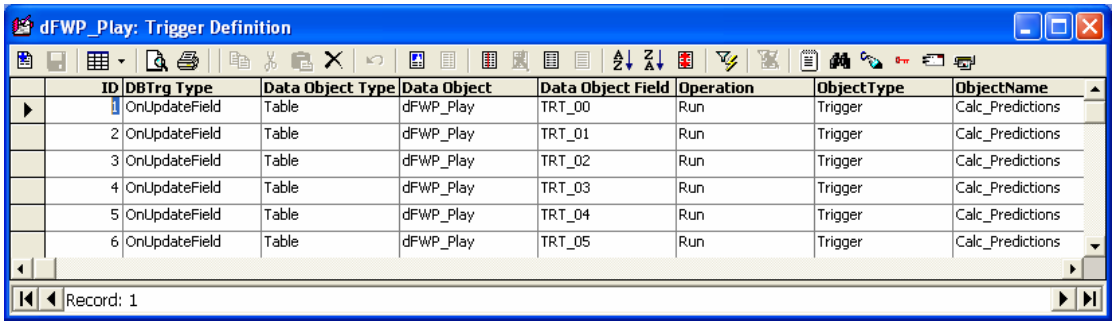
17.1 Define Trigger for Data Object

Trigger is dependent object attached to data objects such as table and query.

Create Trigger

Login as NODEM configurer and select Data Administration mode, select a table, then select  to define trigger for the data object.





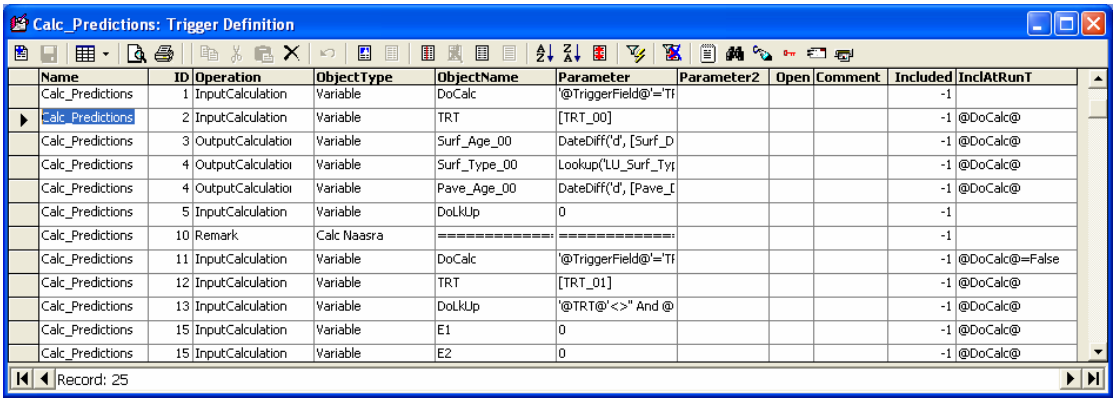
Columns for Object Trigger are:

- **ID**: identifier of a trigger
- **DBTrgType**: NODEM trigger type. Currently OnUpdateField and OnUpdateRecord

- **DataObjectType**: NODME object type, such as table, query etc
- **DataObject**: NODEM object name
- **DataObjectField**: field name in data object
- **Operation**: operation to be called when field value change
- **ObjectType**: type of object to be called for operation. One type is Trigger, and here is the only entry for defining trigger detail
- **ObjectName**: object to be operated by operation
- **Parameter**: parameter applied when calling an operation, normally a filter string and useful for OnRecordUpdate
- **Open**: open to display/edit object detail
- **Comment**: comment
- **Included**: included or not at runtime

17.2 Define Trigger Detail

Many triggers can be defined for the fields of one data object. And in turn one field can have many trigger operation as well. If the object type is Trigger, then the trigger detail can only be defined from here by clicking button displayed in **Open** field.



Name	ID	Operation	ObjectType	ObjectName	Parameter	Parameter2	Open	Comment	Included	InclAtRunT
Calc_Predictions	1	InputCalculation	Variable	DoCalc	'@TriggerField@'='Tf				-1	
Calc_Predictions	2	InputCalculation	Variable	TRT	[TRT_00]				-1	@DoCalc@
Calc_Predictions	3	OutputCalculatio	Variable	Surf_Age_00	DateDiff('d', [Surf_D				-1	@DoCalc@
Calc_Predictions	4	OutputCalculatio	Variable	Surf_Type_00	Lookup('LU_Surf_Typ				-1	@DoCalc@
Calc_Predictions	4	OutputCalculatio	Variable	Pave_Age_00	DateDiff('d', [Pave_[				-1	@DoCalc@
Calc_Predictions	5	InputCalculation	Variable	DoLkUp	0				-1	
Calc_Predictions	10	Remark	Calc Naasra	==========	==========				-1	
Calc_Predictions	11	InputCalculation	Variable	DoCalc	'@TriggerField@'='Tf				-1	@DoCalc@=False
Calc_Predictions	12	InputCalculation	Variable	TRT	[TRT_01]				-1	@DoCalc@
Calc_Predictions	13	InputCalculation	Variable	DoLkUp	'@TRT@'<>'' And @				-1	@DoCalc@
Calc_Predictions	15	InputCalculation	Variable	E1	0				-1	@DoCalc@
Calc_Predictions	15	InputCalculation	Variable	E2	0				-1	@DoCalc@

Columns for defining trigger details are:

- **Name**: trigger object name
- **ID**: trigger execution order
- **Operation**: list of operation supported
- **ObjectType**: object type for the operation
- **ObjectName**: object name for the operation
- **Parameter**: parameter for the object operation
- **Parameter2**: second parameter for the object operation
- **Open**: a quick linkage to display/define an object
- **Comment**: comment
- **Included**: included or not when load all definitions
- **InclAtRunT**: included at runtime or not. Depending on a runtime calculation. Default to true.

Trigger detail definition is very much similar to the definition of other operational objects like batch, model etc with minor difference: only runtime (directive) variables can be used and no compiling supplied for calculation. Everything is done on the fly.

17.3 Use Trigger

There is no direct calling to execute a trigger – as it is basically tied up with data change. The running of a trigger is automatically fired when a data change happens in a data object with trigger defined.

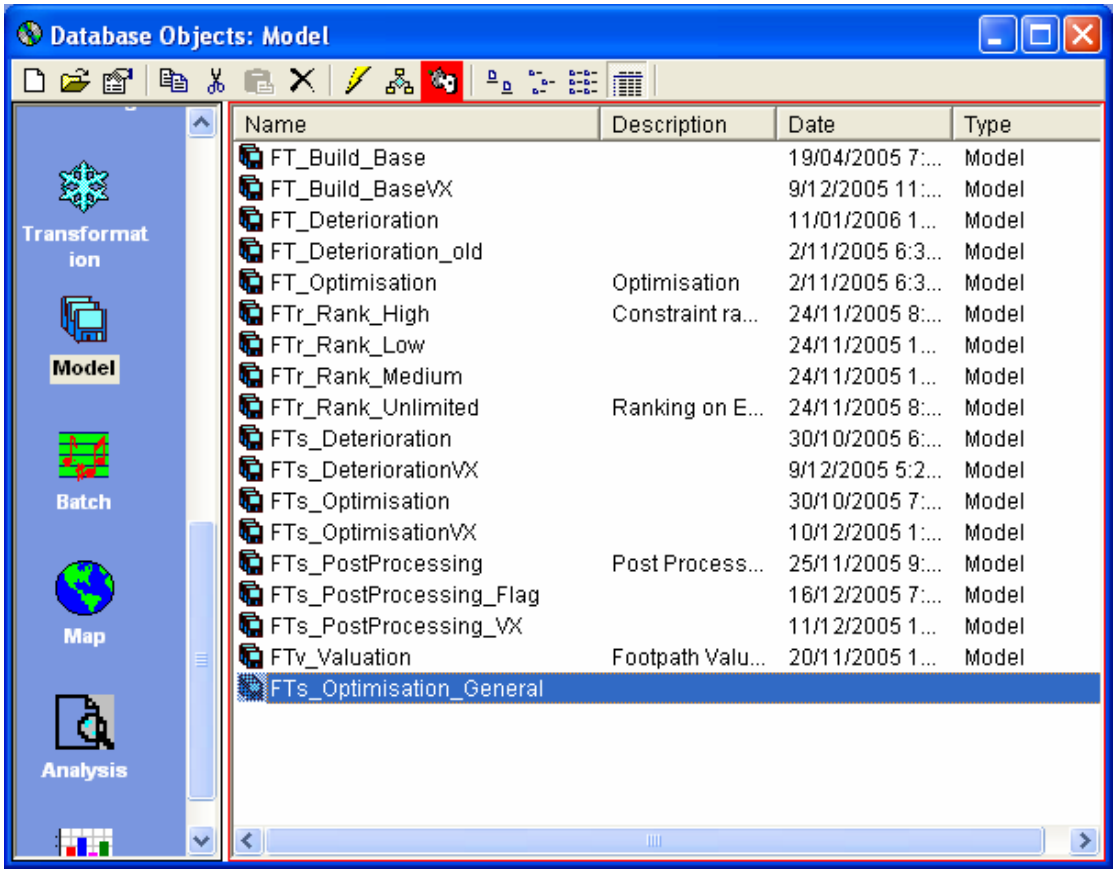
18 WORKING WITH INFO WIZARDS

Info Wizard is a function to collection preset value for input runtime variables. It is particularly useful for customisation of general definitions of models and batches. Info Wizard is only available for system configure during system development stage.

18.1 Use Info Wizards to Collect Preset Value for Runtime Variables

Start Up Info Wizard

Login as NODEM configurer and select Data Administration mode, then start database object browser. Locate an operational object, the Info Wizard button will be available.



Click  to start Info Wizard, a welcome screen will display.

18.2 Info Wizard Buttons Functions

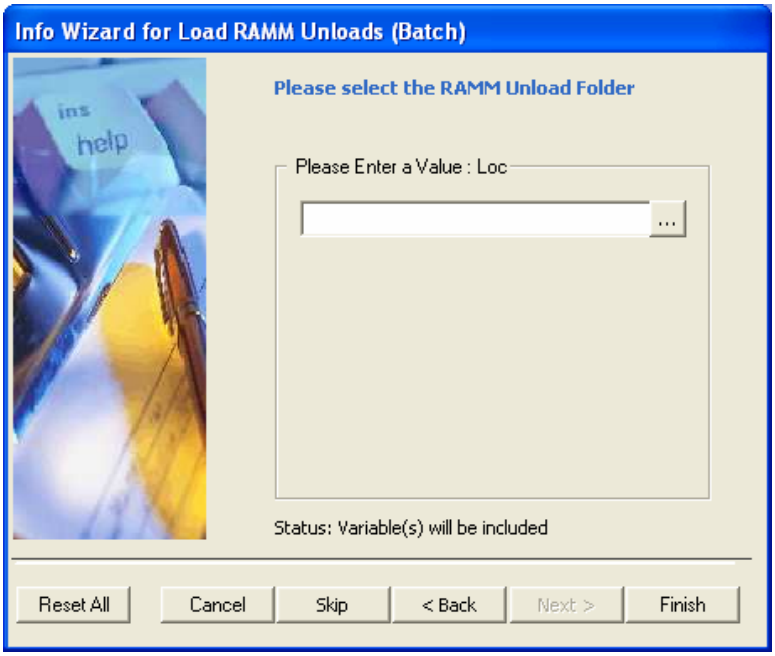
Following are detailed operations for all buttons.

Button	Operation Description
Reset All	Discard all collected values for all variables, and upon save (Finish) all collected (or previously saved) variable values will be thrown away
Cancel	Cancel the operation and newly collected variable values will not be ignored
Skip	Skip the current variable (do not collect value for this variable, or ignore the value if collected). This button displays as **Skip** when the variable has not been collected at all, and **Reset** if value has been collected. It will display **UnSkip** upon click
Reset	Clear and skip the current variable (clear any value collected and skip the current variable)
UnSkip	Value collected for current variable will be kept
< Back	Move to previous input variables. It will be disabled at start and for the first variable. Before you can move to previous variable, you must have a valid input for current variable unless you skip it.
Next >	Start the collection or move to next input variables. It will be disabled for the last variable. Before you can move to next variable, you must have a valid input for current variable unless you skip it
Finish	Fnish the collection operation and save all collected value for the object

18.3 Info Wizard Input Screens

In compliance with the types of runtime input variables, different user interface is provided for variable value collection. Following are a list of input screens.

18.4 Input Text Box (with or without button)

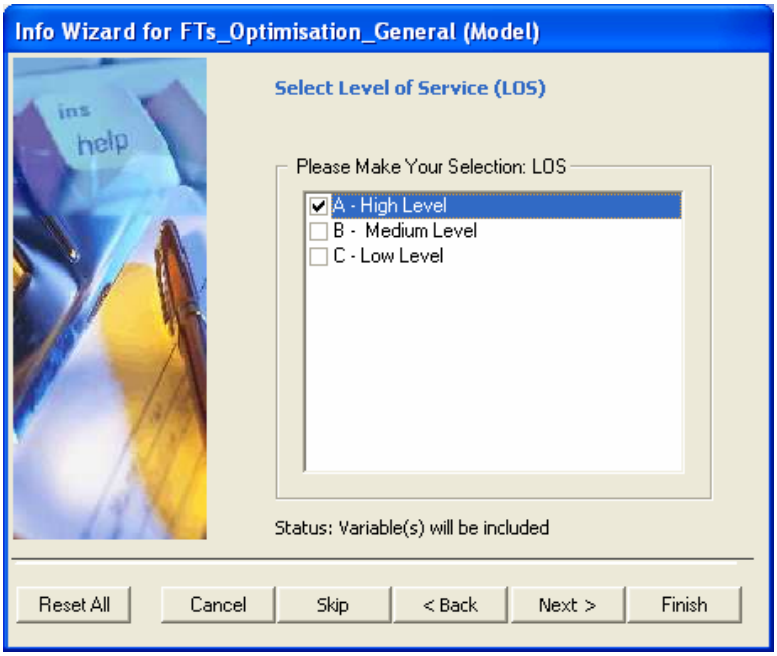


Text box can accept user input directly, although text box with button (for variable types of InputFileFolder, Input FileSave and InputFileOpen) can get the input from corresponding folder browser and file open/save dialog.

- **Input Combo Box**

Input combo box can accept direct user input (such as a simple user input, or a table doest not yet exist but will be created after an object execution), or select from a list of options. The available selection list can be a list of values (simple value, or SQL select string), a list of objects (such as tables), or a list of fields from a table or query.

- **Input List Box**



Input list box can accept input from a list of selections. The available selection list can be a list of values (simple value, or SQL select string), a list of objects (such as tables), or a list of fields from a table or query.

- **Input Check Box**

Input check box collects values for a number of variables, and the value is basically 0 or 1 (or false or true).

18.5 Use Preset Value for Runtime Variables

Preset value for runtime variables is for the purpose of general object configuration and customisation. The value collected for variables can be used once for all.

To use preset values for variables at runtime is simple. Add following operation in the corresponding object before any input runtime variables (this is dependent on the execution order, ie ID field). All other fields will be left blank.

ID	Operation
0	LoadRunTimeVariables

During runtime, any input variable (except InputConfirmation variable) without a preset value will b popped up for a user input. But if a preset value does exist, then that value will be used for that variable throughout the execution period, although the user can overwrite that value in input or output calculation variables.

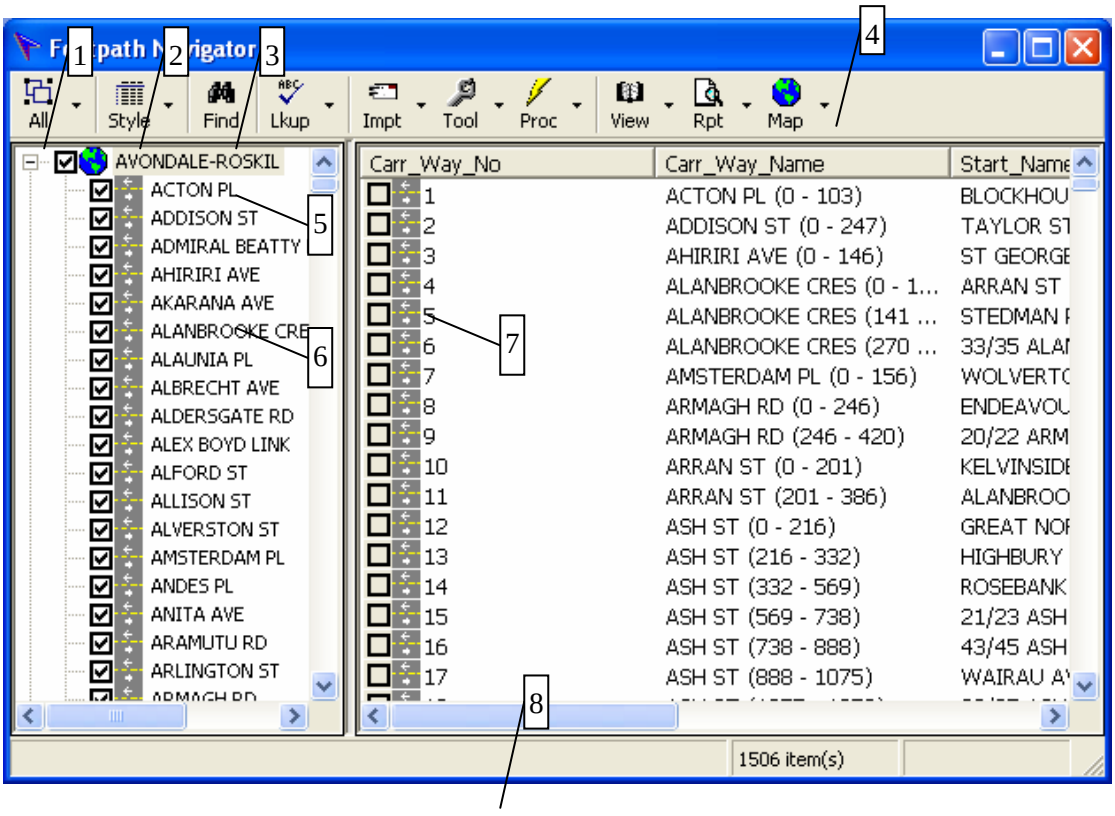
19 WORKING WITH SUB-SYSTEM CONFIGURATION

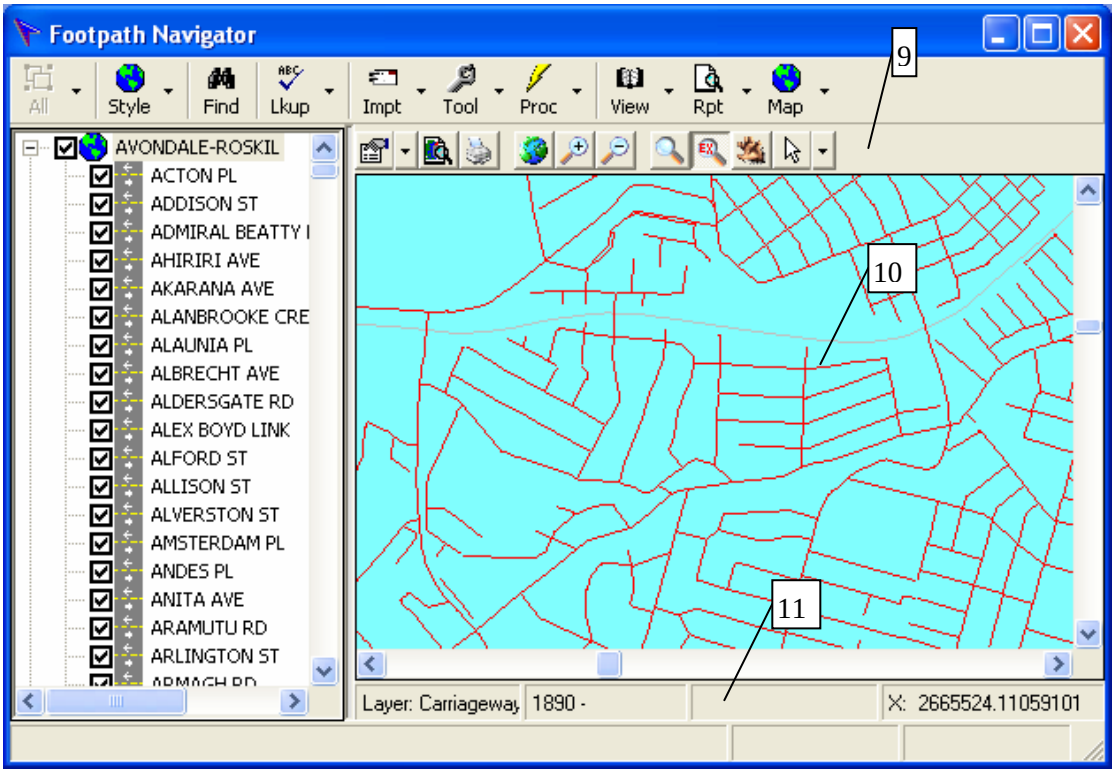
Sub-system is an integrated organisation of similar or related data, functionality, operations and processes to fulfil business requirements from a system point of view. A sub system is one part of or a whole complex business system, but technically itself is a software system from physical or logic perspectives.

Take road network for instance, one road information management system (RIMS) may do all the work. Alternatively, there can be road information system (RIS) to store road data, pavement management system (PMS) to use data for management and administration, a footpath management system (FMS) for footpath data storage and maintenance management etc. All these are sub systems of road network, but actually each in turn is a system in nature by itself.

Sub systems can be easily configured in NODEM with all data and operational objects and presented to the user as a workspace (the main user interface - Navigator). With built-in data navigation, filtering and system wise user access control, sub system navigators make sub systems fully self-contained, consistent and very much user friendly.

Following are a typical sub system navigator





Components can be used or configured inside sub system are

1. *Data Group*: high level grouping of data, such as Hierarchy of road. The data list based on the group will be added under the button for selection.
2. *Style*: detailed data item list view style, including Large Icons, Small Icons, List and Detail for data list, or GIS view if configuration is available.
3. *Find*: a button to find the listed item based on certain criteria. If item is listed in the list view, the found one will be selected; otherwise, no item will be marked
4. *Toolbar buttons for configuration*: a list of operations can be configured under different buttons, including Lookup, Setup, List, Import, Tools, Process, Edit Data, View Data, Report, Chart, Thematic Map
5. *Second Level Data Group*: a second level grouping of data, such as Area or Suburb for road.
6. *Data List*: a list of data in parent level, such as road or link
7. *Data Item List*: a list of data items, such as link under road, or sections under road or link etc. All selection of data items are done in this view. Normally if no item listed, then presume all data items; if some items are listed by none are selected (checked), then all listed items is presumed to be included; if some items are selected in the view, then only selected items will be included as selection. Popup menu can be configured to for information or operation as well. Here can also be GIS view, similar law applies
8. *Status bar*: information about listed data items or map information in GIS view.
9. *GIS View Toolbar*: toolbar for GIS view data navigation
10. *GIS View*: basic layers for navigation and selection
11. *GIS Status bar: information about GIS layer*

19.1 Decide Needs for Sub System (Navigator)

Since sub system (sometimes it is simply referred as Navigator) is a system for certain business system, rules of business requirements to a information system and methodologies of software development life cycle (SDLC) applies. Specific to NODEM as a

system platform, a transformational way of thinking and design will need to be maintained in the system development life.

Following is brief guidelines for NODEM sub system configuration (development):

Data Needs
Business data to be captured, integrated or linked from other external data source and data to be processed and presented etc will need to be identified, defined and developed if required. This mainly includes data table, query, various types of documents (such as media, design), GIS data etc. All these can be referred to as data objects inside NODEM.

Process Needs
Business process is business functions and activities exerted on business data. The interaction of activity and data can be implemented as different types of operational objects in NODEM, including trigger (some complex business rules), batch (a bunch of operations), model (data processing and calculation) and analysis (an step by step organisation of operations) etc.

Presentation Needs
Presentation of data refers to user interface and data presentation, including data table/query opening formats, data forms, data reports and graphs, documents such as media, stored charts and thematic maps, and mainly the organisation of all these objects and operational objects for the user to interface with.

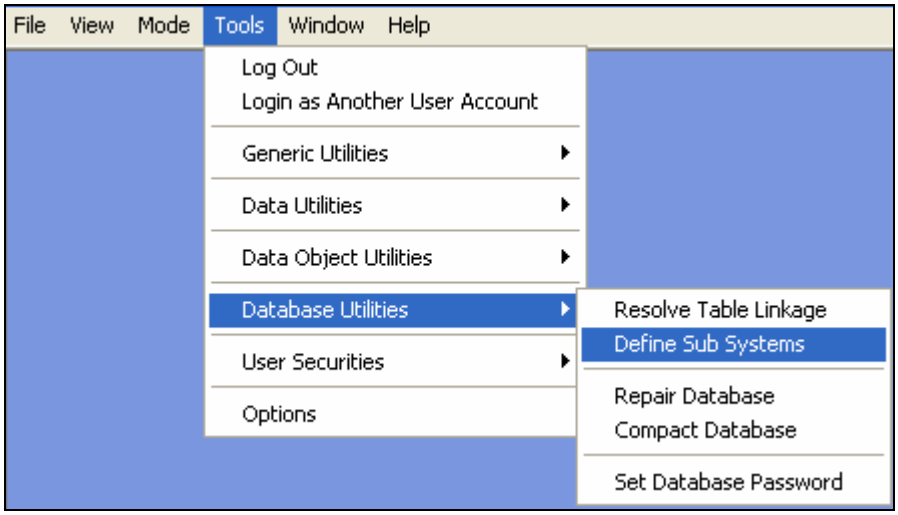
Miscellaneous Needs
Other needs for sub system including consideration for user access, data navigation and filtering and operation integration inside navigator.

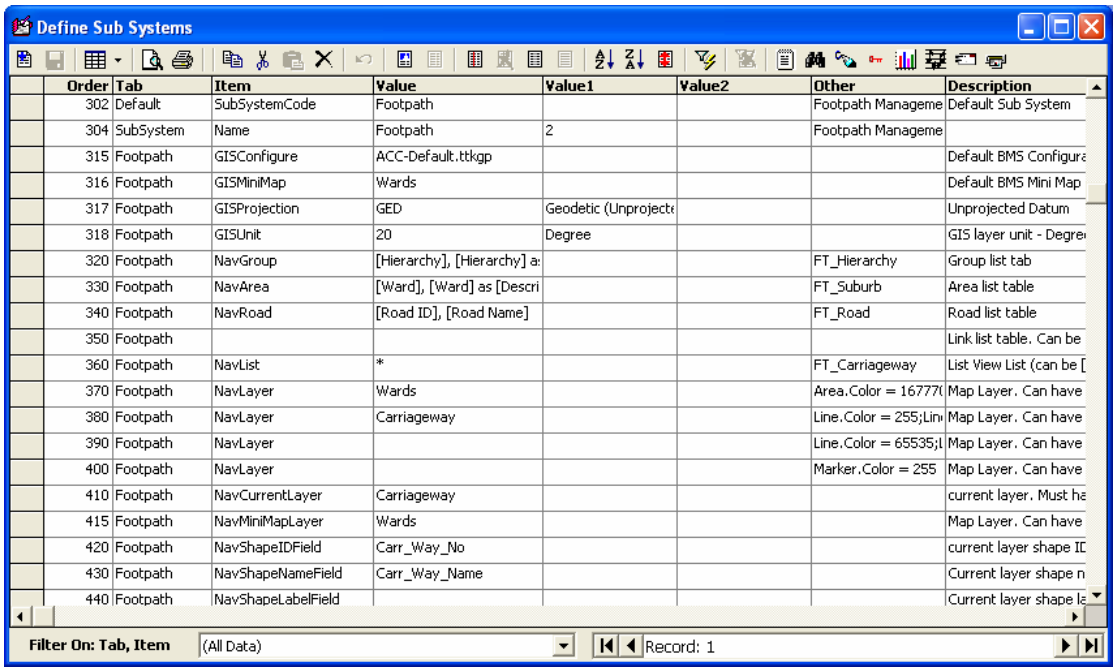
One main part of sub system configuration is about the data navigation, that is, how a system will be most effectively working in terms of business data. There is no simple rule for this, but the client of the system definitely knows better what they want.

19.2 Configure Sub System

Once all data and operation objects are clearly identified and defined (see each section above for detail regarding object definition and design), configuration of sub system is just another type of data entry.

Login as NODEM configurer and select Data Administration mode, then select Tools | Database Utilities | Define Sub Systems to start up sub system definition (yet a sophisticated sub system configuration interface is under development).





19.3 Columns within Sub System Configuration

Columns inside the above data sheets are:

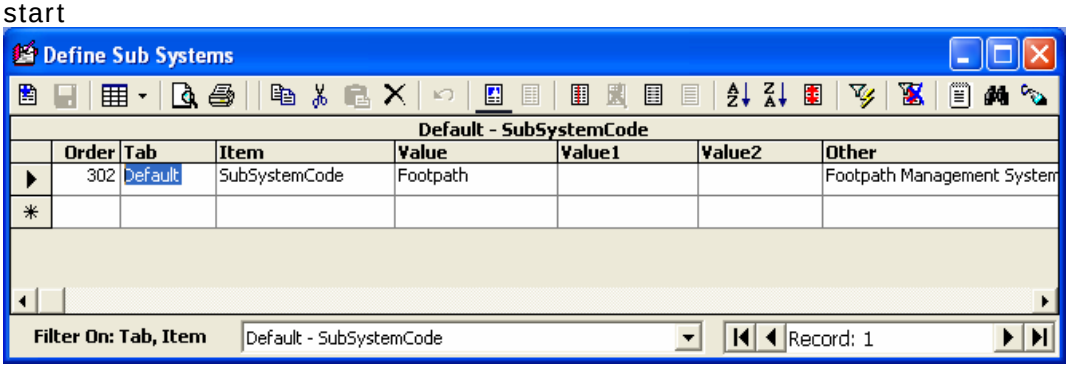
- **Order**: the definition order of system (subsystem(s)). **The order is very important for building subsystem navigator at runtime**
- **Tab**: a sub system group or some global definitions
- **Item**: a sub system entry or some global definitions fro that sub system
- **Value**: the key value for the subsystem entry, normally is object name
- **Value1**: the second value for the subsystem entry, normally is key operation
- **Value2**: the third value for the subsystem entry, normally menu group definition
- **Other**: the fourth value for the subsystem entry, normally caption for display
- **Description**: used for configuration comment
- **Display**: reserved for future use
- **Date**: the time when the entry added

Built-in dynamic filter can be applied on **Tab** and **Item** fields for easy data entry. Some preliminary entries will be defined for a new system to start.

19.4 Combinations of Tab and Item Columns

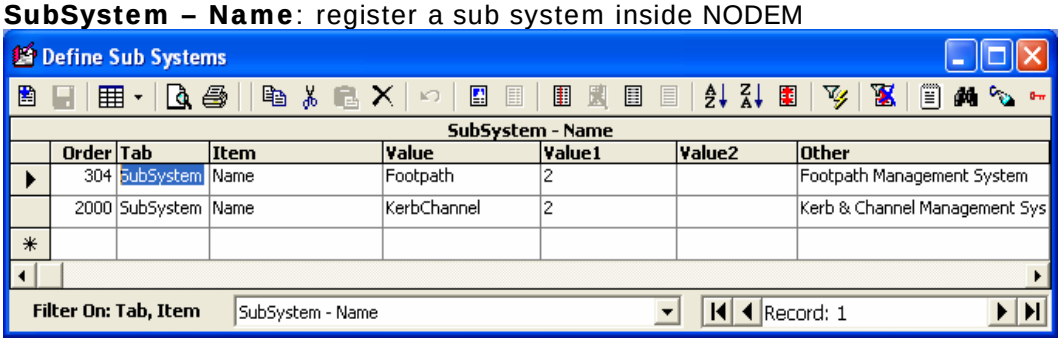
Typical combinations of the values for Tab and Item are:

- **Default – SubSystemCode**: define the default sub system to start when NODEM start



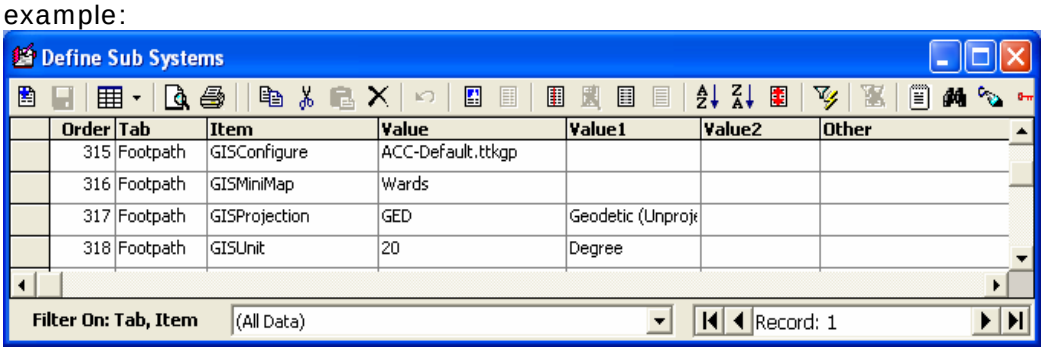
Here *Default* and *SubSystemCode* must be kept, while Value is the code of the system in sub system definition.

- **SubSystem – Name**: register a sub system inside NODEM



Here *SubSystem* and *Name* must be kept, while **Value** will hold the sub system code with **Other** the sub system name (description).

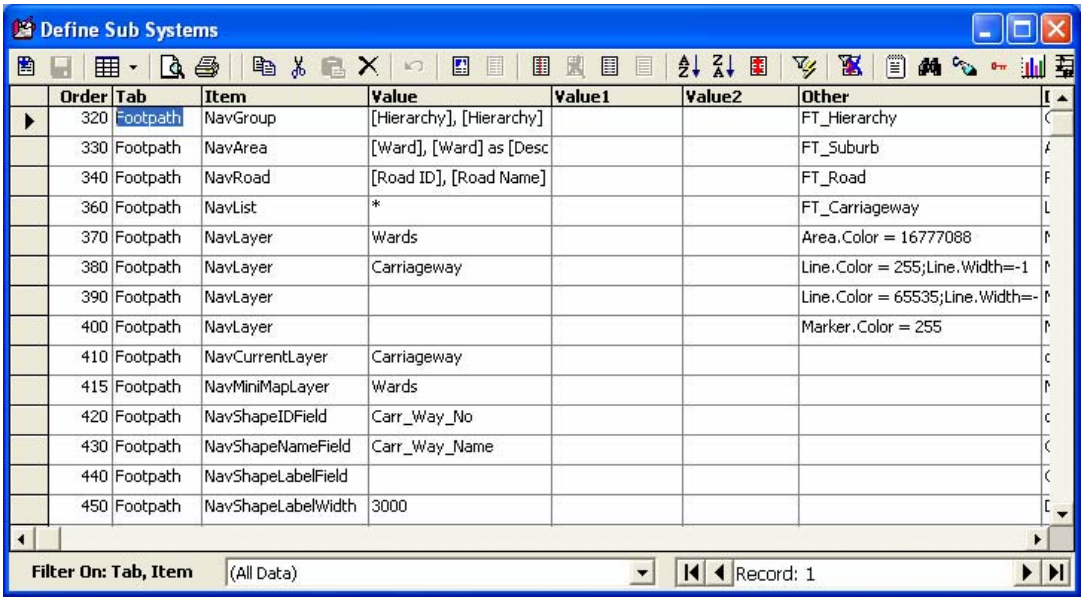
- **Some Sub System Global Definition**: applies to the sub system. They are for example:



  - **Footpath – GISConfigure**: default GIS configurations when any thematic map loaded for the sub system
  - **Footpath – GISMiniMap**: default mini amp layer when any thematic map loaded for the sub system
  - **Footpaht – GISProjection**: GIS map projection
  - **Footpath – GISUnit**: map unit

- **Footpath –Spatial**: map layer included in sub system (can have popup menu definition defined in Value2 as Field, Caption, Operation;…)
- **Footpath – Popup**: popup menu definition in navigator

Sub System Navigation Definition: for sub system data navigation, including a list of combinations in a format of Tab as sub system code with item starting with "Nav". They are for example:



Order	Tab	Item	Value	Value1	Value2	Other
320	Footpath	NavGroup	[Hierarchy], [Hierarchy]			FT_Hierarchy
330	Footpath	NavArea	[Ward], [Ward] as [Desc			FT_Suburb
340	Footpath	NavRoad	[Road ID], [Road Name]			FT_Road
360	Footpath	NavList	*			FT_Carriageway
370	Footpath	NavLayer	Wards			Area.Color = 16777088
380	Footpath	NavLayer	Carriageway			Line.Color = 255;Line.Width=-1
390	Footpath	NavLayer				Line.Color = 65535;Line.Width=-
400	Footpath	NavLayer				Marker.Color = 255
410	Footpath	NavCurrentLayer	Carriageway			
415	Footpath	NavMiniMapLayer	Wards			
420	Footpath	NavShapeIDField	Carr_Way_No			
430	Footpath	NavShapeNameField	Carr_Way_Name			
440	Footpath	NavShapeLabelField				
450	Footpath	NavShapeLabelWidth	3000			

- **FoothPath – NavGroup**: data definition for navigation data group (here based on hierarchy)
- **Footpath – NavArea**: data definition for navigation area for second level of grouping (here ward)
- **Footpath – NavRoad**: data for navigation data list or third level of grouping (here Road) This can be SubSystemCode – Link, or under Footpath –Road.
- **Footpath – NavList**: data for navigation data item (here foothpath carriageway section)
- **Footpath – NavLayer**: GIS layer((s) for navigation
- **Footpath – NavCurrentLayer**: GIS layer for all navigation operations
- **Footpath – NavMinMapLayer**: GIS mini map layer for area location
- **Footpath – NavShapeIDField**: used for both GIS data and list view data
- **Footpath – NavShapeNameField**: used for both GIS data and list view data
- **Footpath – NavShapeLabelField**: used for GIS display
- **Footpath – NavShapeLabelWidth: used for GIS display.**

- **Sub System Toolbar Definition**: configuration of all toolbar buttons. The combinations can be:
  - **Footpath – Lookup**: define all lookup data for analysis
  - **Footpath – List**: define all lookup list for data entry
  - **Footpath – Setup**: define all set up for analysis
  - **Footpath – Import**: define data import functions
  - **Footpath – Tools**: define some functions or utilities to include
  - **Footpath – ExecProc**: define running processes to include
  - **Footpath – EditData**: define data to edit
  - **Footpath – ViewData**: define data to view
  - **Footpath – Report**: define reports to include

- **Footpath – Chart**: define stored charts to include
- **Footpath – SpatialView**: define thematic map to include

Default access levels for these buttons are:

Function	Access Level	Access Mode	Modify Level	Modify Mode
Lookup	All	All	PowerUser	Operation
List	All	All	Admin	Administration
Setup	All	All	Analyst	Operation
Import	Power User	Operation	Not applicable	
Tools	Power User	Operation		
Process	Power User	Operation		
EditData	Editor	All	Editor	All
ViewData	All	All	Not allowed	Not allowed
Report	All	All	Default object access control applies	
Chart	All	All		
SpatialView	All	All		

Extension for the access control can be done through redirection. For instance, if a chart is only supposed to operate by analyst, then we can add it under ***Tools*** but execute it through ***Chart*** function, the format is Value->Chart.

19.5 Typical Values for Other Fields

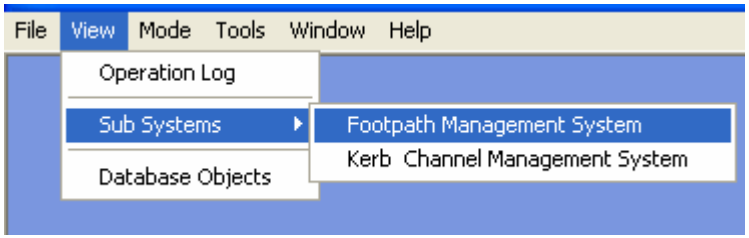
Combination of Tab and Item values describe most of the functionalities existing NODEM sub system, the detailed operations and objects will need to be defined. Following is the possible objects for different tool bar buttons.

Function	Value	Value1	Value2
Lookup	Table, query, form name[1]	[2], Open	[2], Group[3], Obj@[Type], Grp@[Group], GroupItem, GroupItem@(Grp@[Group]Obj@[Type])
List	Table, query, form name	, Open	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem@(Grp@[Group]Obj@[Type])
Setup	Table, query, form name	, Open	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem@(Grp@[Group]Obj@[Type])
Import	Table name		
Tools	Any type of object, CMT, MRD[4]	Data or operational object operations (such as Open, Run, w/o filter etc)	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem(@Grp@[Group]Obj@[Type]). Anything extra will be filtering for selection list build. Data feeder group name for CMT (commit) and MRD (most recent) operation
Process	Operational object name	, Run, RunOrRunFilter, RunFilterOnly, RunOrRunFilter@[Field](as[Alias]), RunFilterOnly@[Field](as[Alias])	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem(@Grp@[Group]Obj@[Type]). Anything extra will be filtering for selection list build
EditData	Table, query, form name	Open, OpenFilterOnly, Filter, OpenFilterOnly@[Field]as[Alias], Filter@[Field]as[Alias]	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem(@Grp@[Group]Obj@[Type]). Anything extra will be filtering for selection list build
ViewData	Table, query, form name	Open, OpenFilterOnly, Filter, OpenFilterOnly@[Field]as[Alias], Filter@[Field]as[Alias]	, Group, Obj@[Type], Grp@[Group], GroupItem, GroupItem(@Grp@[Group]Obj@[Type]). Anything extra will be filtering for selection list build
Report	Report	, Open (OpenFilterOnly and Filter not supported yet)	, Group, GroupItem, Obj@[Report], GroupItem@Obj@[Report]
Chart	Chart name	Run	, Group*, GroupItem, Obj@[Chart], GroupItem@Obj@[Report]
SpatialView	Thematic map	Open	
Data Feeder	Table name		Feeder (used to build a list of tables for carrying out committing data and flagging most recent data)
Popup	FieldName	, Display Image (Video), Display Data Table	

[1]: an object can be defined under one function group (button), but executed through another function group. This is called redirection. The format is ObjectName->Function Group, for instance, defining a chart object FWP_Chart under Tools but run through chart, FWP_Chart->Chart
[2]: value can be empty, and default operation will be used
[3]: when a Group is used, a menu group with sub menu will be created. All submenu are defined by GroupItem
[4]: CMT and MRD must be under tools and are internal name for dedicated functions. A list of data tables (Feeder) will be defined for the operation.

19.6 Use Sub System

Sub system(s) definition is embedded within database. Once a sub system is defined, it will be automatically added to NODEM system menu under View | Sub Systems. The one set as default will automatically start. A sub system can also be started by selecting from the sub menu.



19.7 System Built-in Statements

Here are some NODEM system built-in statements

Runtime input statement		Applies to batch, trigger, model etc
	Operation	description
	Comment or Remark	anything
	LoopStart*	applies to trigger only. Issue edit data command
	UnAttended	execution cannot be stopped
	NullStringValue	default value if Null
	NullNumberValue	default value if Null
	NullDateValue	default value if Null
	InputTextBox	gain input from user
	InputComboBox	gain input from user
	InputListBox	gain input from user
	InputCheckBox	gain input from user
	InputCalculation	calculate a variable expression
	OutputCalculation	calculate a variable expression and save it to database. Applies to trigger only
	InputFileFolder	browse for a folder as input
	InputFileOpen	browse for a file to open
	InputFileSave	browse for a file to save

	InputConfirmation	message for confirmation (continue or not etc)
	LookupData*	lookup data for runtime variables
	ShowMessage	display message (or status or variable value etc) in operation log (if Variable, then value; if Confirmation, then as confirmation)
	RunCommand	Reserved
	RunSQL	run SQL statement
	LoopEnd*	Applies to trigger only. Issue update command and resume to LoopStart
	Anything else	Treat as object operation (such open table, run mode, run chart etc)
	* Item defined in model works differently and not for runtime variables	
Model Statement	Including model definition and runtime controlling	except Runtime Variables, all will only be resolved at model execution time
general statement		
	InputTextBox, InputComboBox, InputListBox, InputCheckBox	runtime variable (before model execution)
	InputCalculation	runtime variable (before model execution)
	UnAttended	whole process
	NullStringValue	whole process
	NullNumberValue	whole process
	NullDateValue	whole process
	Remark, Comment	comment only
	ShowMessage	Variable, Constant or Confirmation, can be used as runtime controlling
	ResumeFromError	whole process
	RunCommand	Reserved
	RunOperation	execute operation (before model execution)
	RunSQL	execute operation (before model execution)
	ProgressControl	display execution progress (Model execution only)
	PreCompileMode	for model execution
	MaximumAlternatives	for model execution (generate strategy only)
	AllowDoNothing	for model execution (generate strategy only)

	BreakOn, SkipOn, FilterOn, IFStart	model runtime logic controlling
	LookupData	model runtime using
	OutputData	model runtime using
	LoopStart	model runtime logic controlling. Can loop through a table, query or a value range
	ClearReset, LoadReset, WriteReset, Reset	model runtime controlling (generate strategy only)
	Intilialise, Calculate, Update, NewUpdate,LoopEnd, ElseStart, IfEnd, Directive	model runtime controlling (generate strategy only)
Dedicated Statement		
	OptOrganiseData	organise strategy data (finish the link list) (part of optmisation)
	OptTransposeData	transpose data to make each strategy has only one record (part of optmisation)
	OptCalcNPVBCR	calculate NPV BCR based on transposed data (part of optmisation)
	OptCalculateIBC	calculate IBC based on transposed data (part of optmisation)
	OptDoOptimisation	do optimisation based on transposed data with NPV or IBC etc (part of optmisation)
	PriDoPrioritisation	Do prioritisation based on data (can be any source)
	PrgSchDoProgramSchedule	do programming and scheduling (data need to be organised first after optimisation or prioritisation)
Optimisation Definition	***Optimsation can include data organising, transposing, BC/NPV/IBC calculation and optimisation***	
	OptimiseDefStart	definition start
	OptSourceData	source data to be optimised (organize data output as well)
	OptMaxIteration	maximum run iteration for optmisation
	OptOutputData	optimisation output (or transpose output as well)
	OptIdentityField, OptAlternativeField, OptPrevAlternativeField, OptStrategyYrField, OptDiscFactorField, OptInflateFactorField	field match definition

OptTrtCostField, OptTrtTypeField, OptTrtCatgField, OptBenefitField	Root definition to do field match (eg field may be a combination of year, type, category etc).
OptPIAreaField	Field to calculate performance index improvement area (compare with base option)
OptDoNothingTrt, OptDoMinimumTrt, OptRoutineTrt, OptStrategyYrLimit, OptFinancialTax	a value (or constant) to assign to these definitions
OptBudgetCategory, OptBudgetScenario, OptAnalysisPeriod,	a value (or constant) to assign to these definitions, and will be used to resolve budget scenario definitions
OptBenefitFactor, OptCostFactor	factor used inside calculation
OptTotDiscCostName, OptTotDiscBenefitName, OptTotUnDiscCostName, OptTotUnDiscBenefitName, OptDiscBCName, OptUnDiscBCName, OptIBCName, OptBCRName, OptNPVName	field for writing total data (and may or may not exist)
OptIBCOnField	Apply IBC calculation on field name
OptIBCCheckBC	Check Benefit>Cost or not
OptPIAreaFieldSlope	1: upward slope;-1: downwards slope
OptSelName	Select field name. Temporarily for optimisation (not used)
OptTransposeYrLmt, OptTransposePrdCostName	Data transpose definition
OptMaxFieldName OptMinFieldName	field to do Max or Min calculation if based on value only. Field name in Output table. Can be benefit or BCR for Max, Field name in Output table. Can be cost or NPV etc for Min
OptOptimiseOnOrder	except IBC, order for other case. If not defined, will use default
OptOptimalRoot	optimal field Root

	OptPickupFirstTrt, OptPickupSecondTr	generate First or Second Treatment related data or not
	OptOnDiscBC	optimisation on Discounted BC (not used)
	OptFinishOnMinCostOpt	Stop if budget not enough for min cost (not used)
	OptIncludeField	Field(s) included for transpose (and it is year based)
	OptFilterField	Field(s) included for transpose (and not year based)
	OptBudgetPeriodData	budget period data definition
	OptBudgetPeriodID, OptBudgetPeriodFrom, OptBudgetPeriodTo	Field definition inside budget period data
	OptimiseDefEnd	definition end
Prioritisation Statement	***Prioritisation only work on calculated result, and data order must be clearly defined in data source.***	
	PriorityDefStart	
	PriAllowBudgetToCarry	allow budget to carry to next year or not
	PriorityData	data as source and output
	PriIdentityField, PriTrtCostField, PriTrtTypeField, PriTrtCatgFieldPriSelName, PriTrtYrFldName	Field match definition
	PriBudgetCategory, PriBudgetScenario	Root definition to do field match (eg field may be a combination of year, type, category etc).
	PriBudgetPeriodData	budget period data definition
	PriBudgetPeriodID, PriBudgetPeriodYear	Budget period field definition (only year, not from and to, so need to organise the original calculation right as well)
	PriorityDefEnd	definition end
Programming/Scheduling	***Before can be played, program/scheduling data must be organised to feed this model***	
	PrgSchDefStart	definition start
	PrgSchDataTable, PrgSchDataQuery	data table or query to play on
	PrgSchBudgetScenario	budget scenario definition
	PrgSchIdentityField, PrgSchSelectField, PrgSchModifyField, PrgSchDataSourceField	field match definition

PrgSchCostAdjField, PrgSchOptSelectedField, PrgSchPriSelectedField, PrgSchTrtTypeField	field match definition
PrgSchTrtYearField, PrgSchTrtCostField, PrgSchTrtCatgField, PrgSchOptTrtYearField	field match definition
PrgSchOptTrtTypeField, PrgSchOptTrtCostField, PrgSchOptTrtCatgField, PrgSchMCAField	field match definition
PrgSchCostData, PrgSchMCAData, PrgSchPriorityData, PrgSchOptimalData	data to be used
PrgSchMCAModel	MCA model to be executed during playing