



# Prototype User Guide

## Public Release Version 1

Related task of the project (Task # and full name):	Task 2.6 - Prototype
Author:	James Carr
Prepared by:	
Approved by:	



## Table of contents

1	Definitions.....	2
1.1	Acronyms and Abbreviations .....	2
2	MDL IDE .....	4
2.1	Workspaces .....	4
2.2	Public Release: Importing Use Cases .....	4
2.3	Quick Start .....	5
2.4	Editors and Views .....	6
2.4.1	Project Explorer .....	6
2.4.2	MDL Editor .....	6
2.4.3	Outline View .....	6
2.4.4	Consoles.....	8
2.5	Projects .....	9
2.5.1	Creating .....	9
2.5.2	Importing .....	10
2.5.3	Structure.....	11
2.5.4	Closing .....	12
2.5.5	Opening .....	12
2.5.6	Deleting .....	12
2.6	MDL .....	14
2.6.1	Creating .....	14
2.6.2	Importing .....	14
2.6.3	Structure.....	17
2.6.4	Errors and Warnings .....	17
2.6.5	Syntax Highlighting .....	18
2.6.6	Auto Completion .....	18
2.6.7	Templates .....	18
2.7	Conversion .....	21
2.7.1	MDL to NM-TRAN.....	21
2.8	Troubleshooting .....	22
2.8.1	The “Run with NONMEM” option fails .....	22
2.8.2	The Check for Updates says no update sites to search .....	22
2.9	Additional Help .....	23
3	Document Revision History .....	24

## 2 MDL IDE

The MDL IDE is the environment in which MDL files can be edited with advanced features such as auto complete and syntax highlighting. If you are familiar with the eclipse platform you will notice a lot of similarities to the MDL IDE.

### 2.1 Workspaces

A workspace is a location on your computer where all of the projects in the MDL IDE are stored.

When you start the MDL IDE you will be prompted for a location of a workspace (unless you set a default), this can either be a folder containing an existing workspace (if you've used the MDL IDE before) or an empty folder.

You can switch workspaces at any time (File -> Switch Workspace) but remember your projects exist inside a workspace so you will only be able to view them when you have the correct workspace selected.

### 2.2 Public Release: Importing Use Cases

There are a set of use cases provided with the public release which are designed to showcase the features of the MDL. To import these use cases into the MDL IDE first you will need to unzip them then follow the instructions in section **2.5.2 Importing**, selecting the folder into which you have unzipped the use cases.

The video "MDL IDE Importing Use Cases" also covers these steps.

## 2.3 Quick Start

If you are in a hurry you can follow these steps to start editing an MDL file quickly:

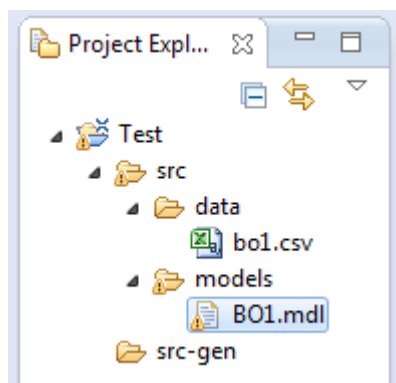
1. Create a project by selecting **File -> New Project**, select **MDL Project** and click next
2. Give it the name TestMDL
3. TestMDL is now visible in the Project Explorer, expand it and find the models folder
4. Right-click on the models folder, select **New -> Other...** and choose **MDL File** then click **Finish**
5. You now have an MDL file, the MDL Editor is open and you're ready to write MDL

The video "MDL IDE Quickstart" also covers these steps.

## 2.4 Editors and Views

### 2.4.1 Project Explorer

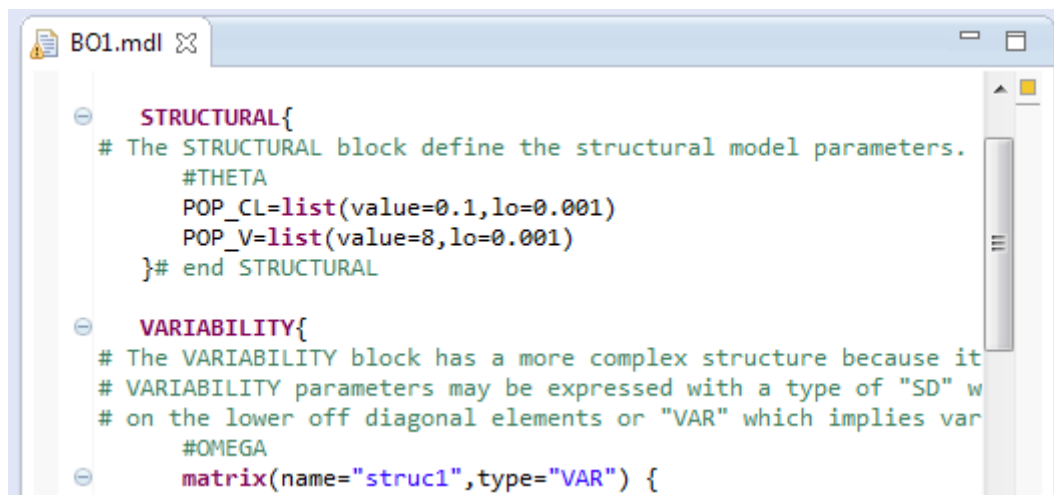
The project explorer lists all of the projects in your current workspace. Projects can be expanded to see the files in a hierarchical tree view. The hierarchy maps to the folder structure held on disk for the project.



Note that closed projects must be opened (see 2.5.5) before they can be interacted with.

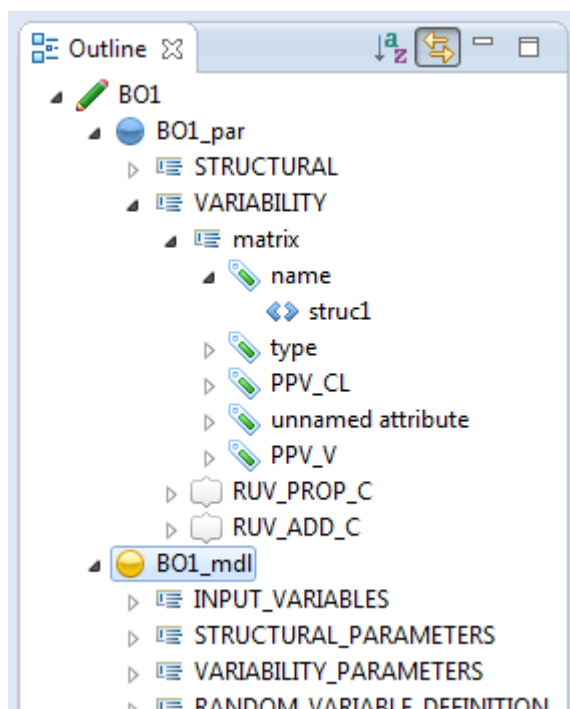
### 2.4.2 MDL Editor

The editor provides the functionality to view and edit MDL files. It is tabbed so you can open multiple MDL files and switch between them quickly.











### 2.4.3 Outline View

When an MDL file is open the outline view will display a hierarchical tree representation of the contents of the file. This is linked to the MDL editor and changes to the file will be reflected here. You can also navigate within the MDL file by selecting an element in the outline view as the corresponding element in the MDL will be highlighted in the MDL Editor.



The elements of the MDL are represented graphically in the Outline view, the following is a legend of the icons used for the elements.

Icon	MDL element
	MDL specification
	Model object
	Parameter object
	Data object
	Task object
	TEL object
	TARGET_CODE block
	MIXTURE block
	Imported function (IMPORT block statement)
	Generic list
	ODE list
	RANDOM list
	ACCEPT instruction
	IGNORE instruction
	DROP instruction / REMOVE list

	ADD list
	Parameter declaration
	Parameter modification
	Reference to a variable or parameter
	Attribute declaration
	Function call
	Expression
	Variable/parameter declaration with transformation, e.g. $\log(x)=0.5$

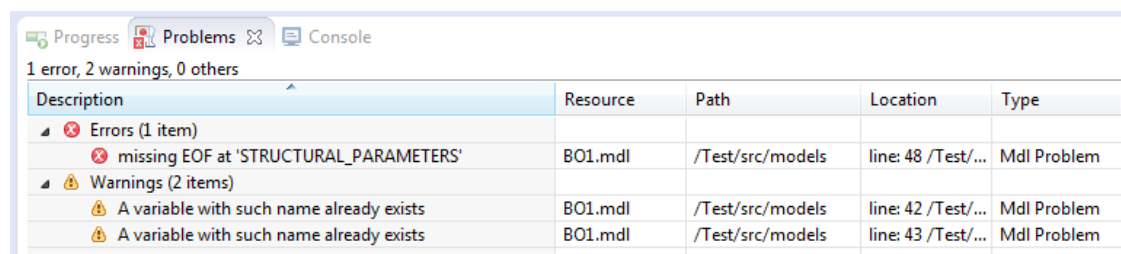
The video “MDL IDE Outline View” gives an overview of how to use the Outline View.

## 2.4.4 Consoles

The consoles section contains three views: progress, problems and console. You can switch between these views by selecting the relevant tab.

### 2.4.4.1 Problems

The problems view shows a list of the problems in the open projects in your workspace. Problems are divided into Errors and Warnings.



Description	Resource	Path	Location	Type
<ul style="list-style-type: none"> <li>Errors (1 item)               <ul style="list-style-type: none"> <li>missing EOF at 'STRUCTURAL_PARAMETERS'</li> </ul> </li> <li>Warnings (2 items)               <ul style="list-style-type: none"> <li>A variable with such name already exists</li> <li>A variable with such name already exists</li> </ul> </li> </ul>	BO1.mdl	/Test/src/models	line: 48 /Test/...	Mdl Problem
	BO1.mdl	/Test/src/models	line: 42 /Test/...	Mdl Problem
	BO1.mdl	/Test/src/models	line: 43 /Test/...	Mdl Problem

### 2.4.4.2 Console

The console view gives a textual output from some of the operations that the MDL IDE performs.

### 2.4.4.3 Progress

The progress view gives a graphical representation of the status of tasks being performed by the MDL IDE as well as providing a cancellation mechanism (red cancel button). Note that not all tasks can be cancelled.



## 2.5 Projects

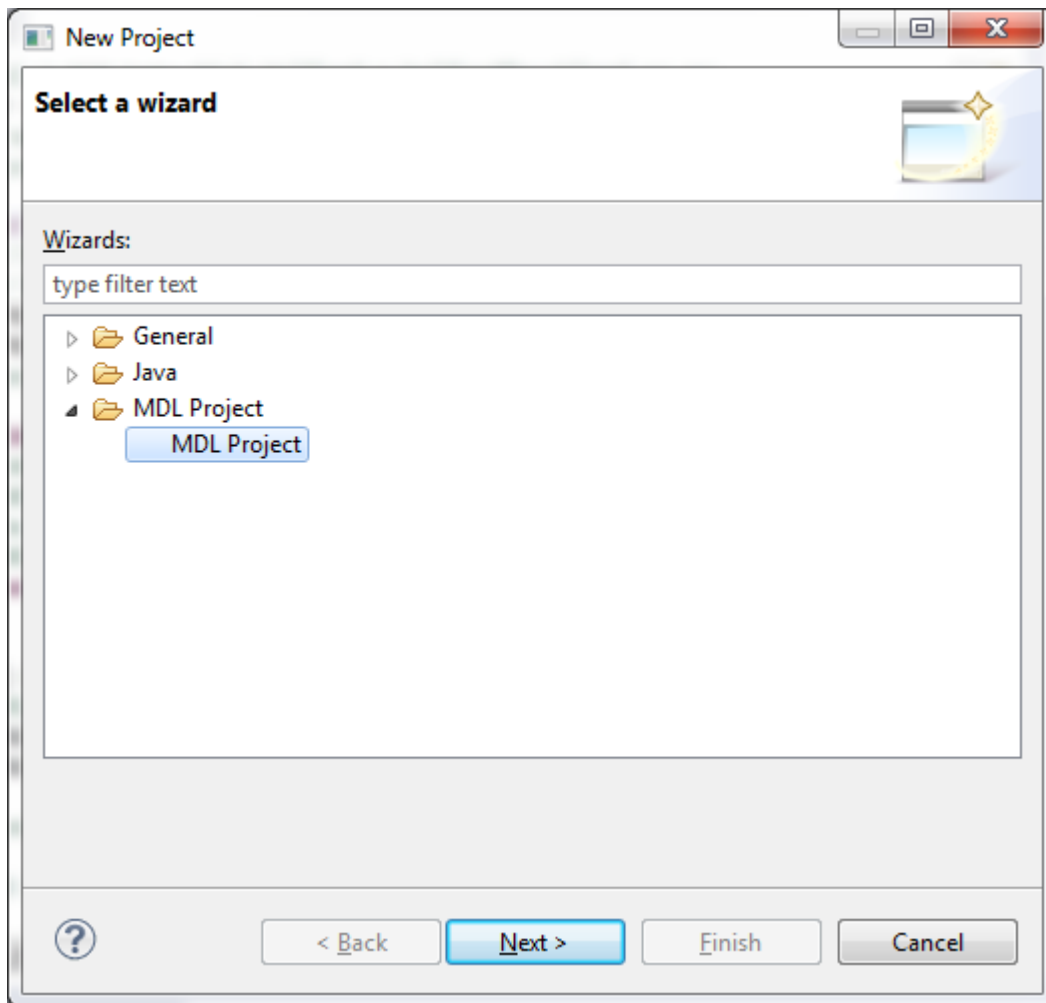
All files in the MDL IDE exist inside a project so you will need at least one project to work on MDL files. You can have as many projects as you want in a workspace, however large numbers of *open* projects will affect the performance of the MDL IDE.

The video “MDL IDE Working with Projects” also covers these steps.

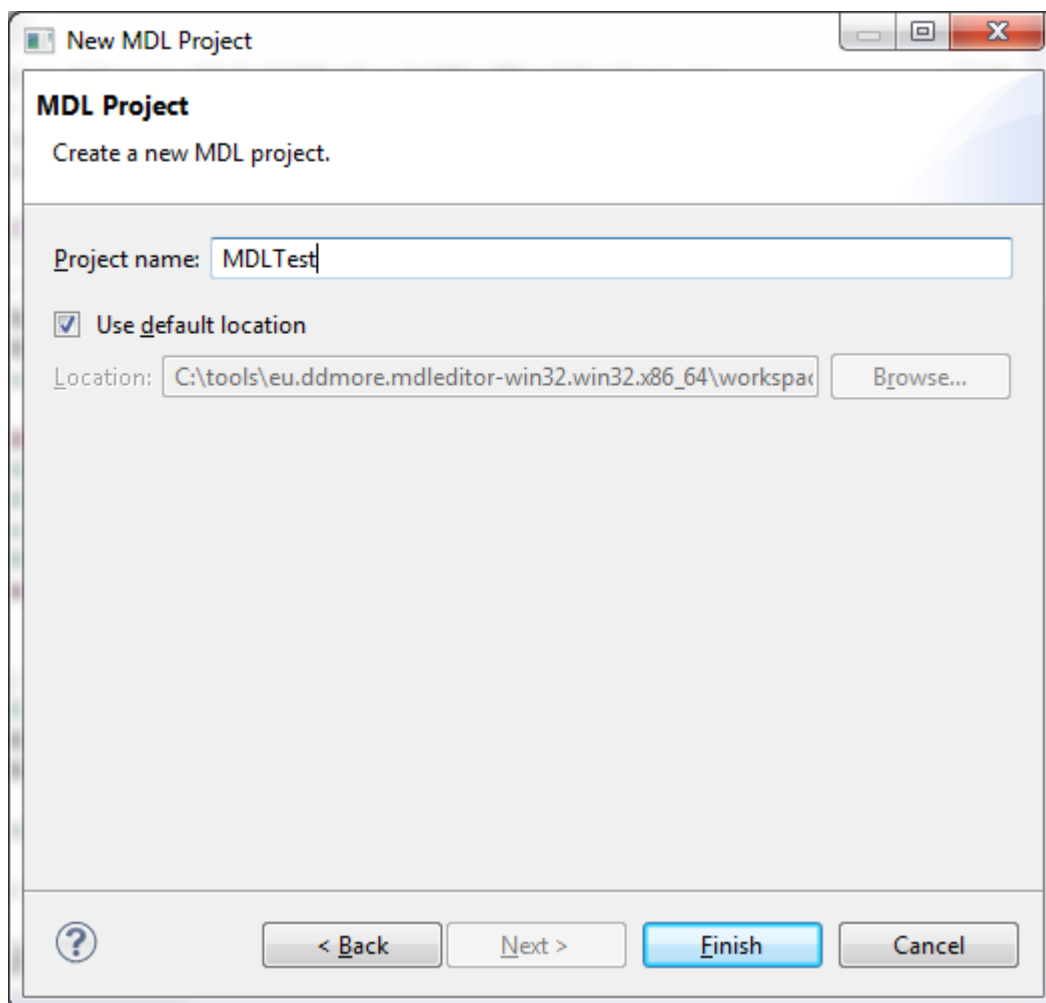
### 2.5.1 Creating

You can create new resources by using the **File > New** menu on the MDL IDE menu bar. Start by creating a simple project as follows:

1. From the menu bar, select **File > New > Project...**



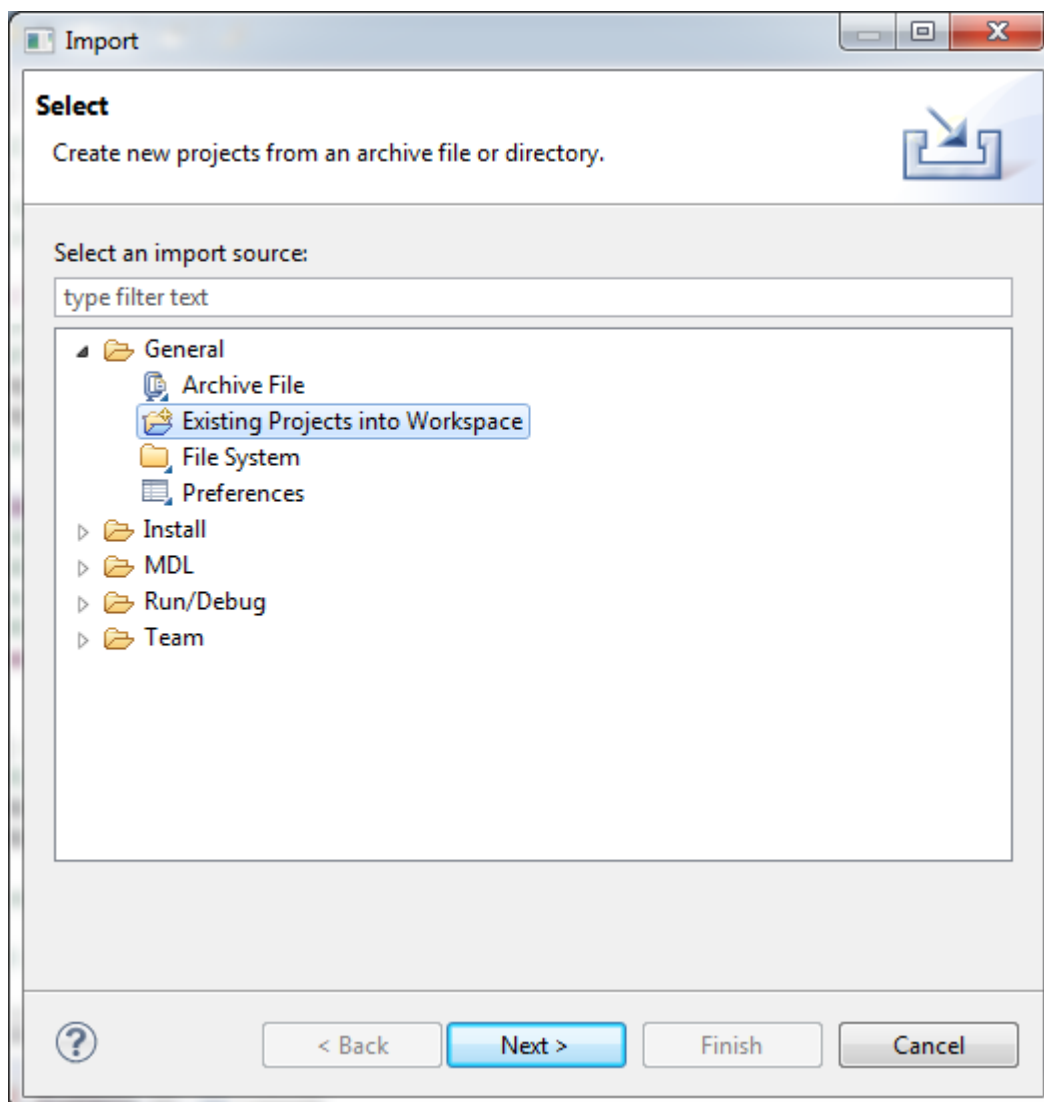
2. In the New Project wizard, select **MDL Project > MDL Project** then click Next.
3. In the Project name field, type your name as the name of your new project, for example "MDLTest".
4. Leave the box checked to use the default location for your new project. Click Finish when you are done.



## 2.5.2 Importing

You can import a project created by the MDL IDE but not currently in your workspace. An example of when you may want to do this is if a colleague has sent you a project they have been working on.

1. From the menu bar, select **File > Import...**

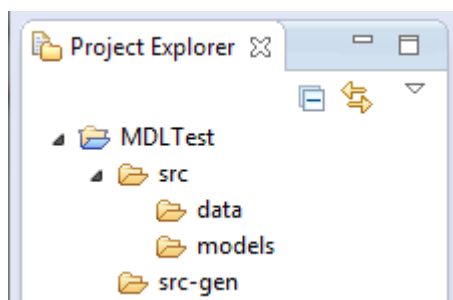


2. In the New Project wizard, select **General > Existing Projects into Workspace** then click Next.
3. Choose either **Select root directory** or **Select archive file** and click the associated **Browse** to locate the directory or file containing the projects.
4. Under **Projects** select the project or projects which you would like to import.
5. Click **Finish** to start the import.

Please note that importing files into a project is a different operation and covered in section 2.6.2 on Importing.

### 2.5.3 Structure

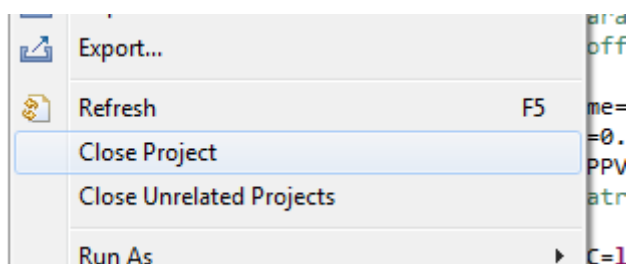
Now that you've created an MDL project you can see in the Package Explorer that some folders have been automatically created for you.



This structure is designed to help you build your project by adding a folder for your models and a folder for data. The **src-gen** folder is a special folder used by the MDL IDE to hold the converted MDL files, when you add and save a valid MDL file in your project this is where the NONMEM control file will appear.

### 2.5.4 Closing

To close a project right click on it and select Close Project.



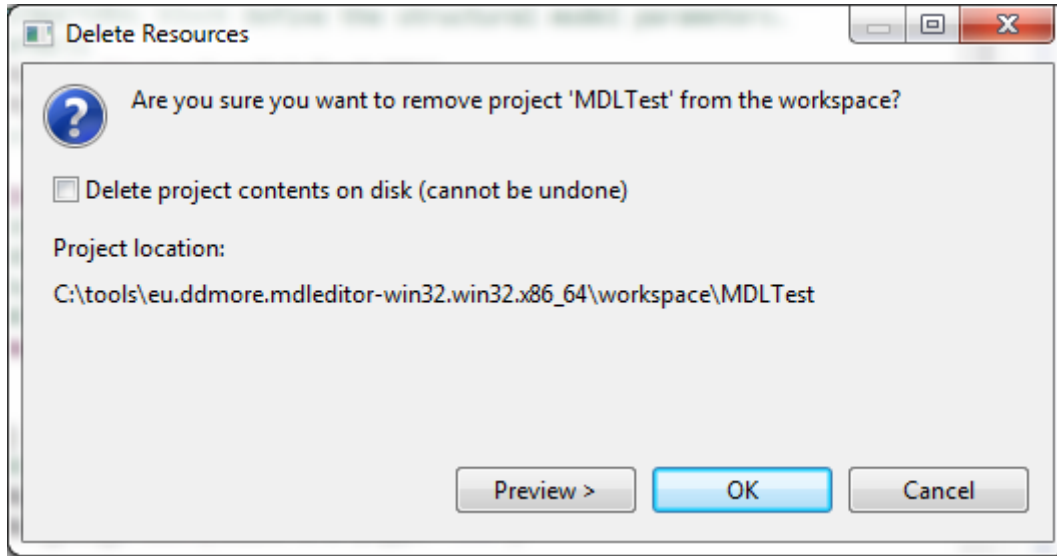
You can close projects that you are not working on currently; this helps to focus the workspace and helps with performance when a number of projects are in the workspace. Closing a project does not remove any information from the project or its files. The project will remain exactly in the same state until you open it again.

### 2.5.5 Opening

You can open a closed project in the workspace by double clicking on it, or with a right click and select Open Project.

### 2.5.6 Deleting

You can remove a project by selecting it and pressing the delete key, or with a right click and select Delete.



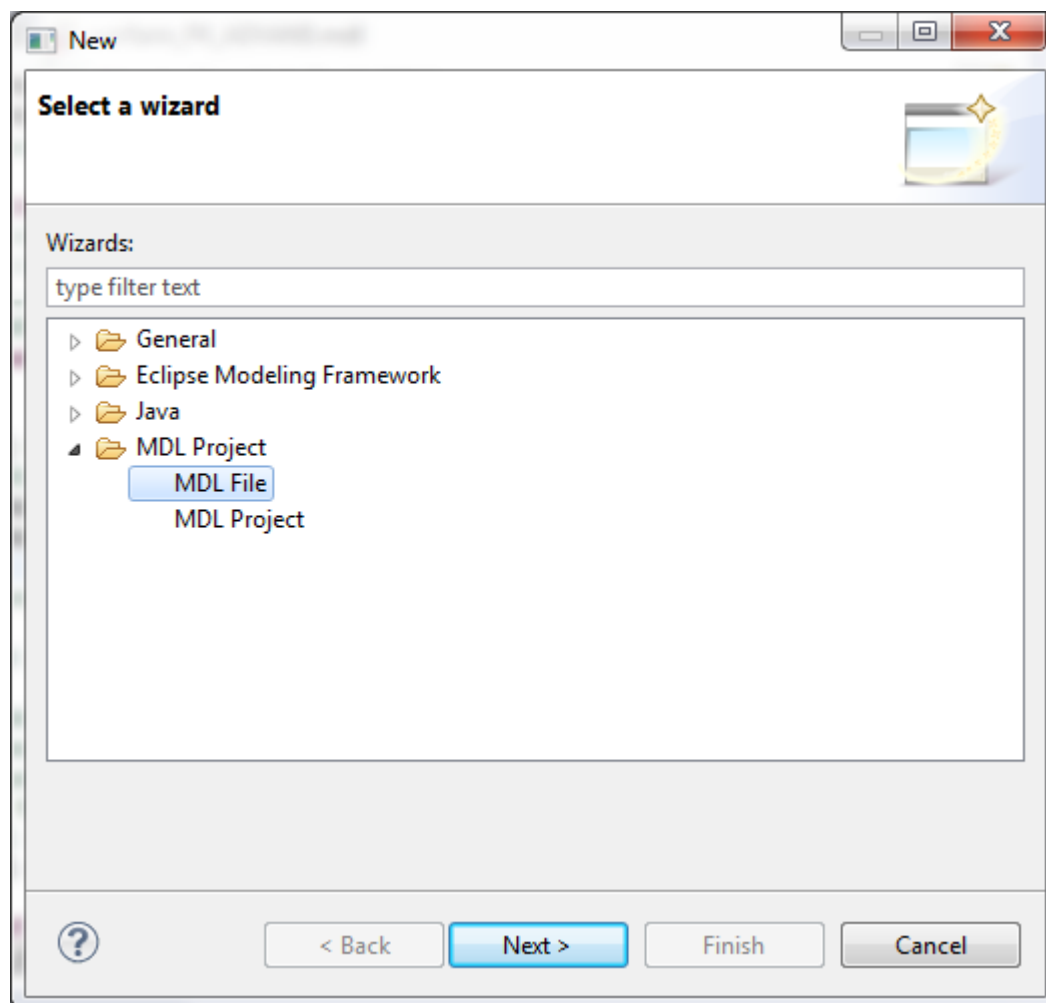
When you attempt to delete a project the MDL IDE will ask you if you also want to “Delete project contents on disk”; if you check this box then the files will be removed from the workspace location and will not be recoverable.

## 2.6 MDL

### 2.6.1 Creating

You can create a new MDL file in a currently open project.

1. Select the folder to which the file will be added in the Project Explorer.
2. Select the **File > New > Other...** menu on the MDL IDE menu bar.



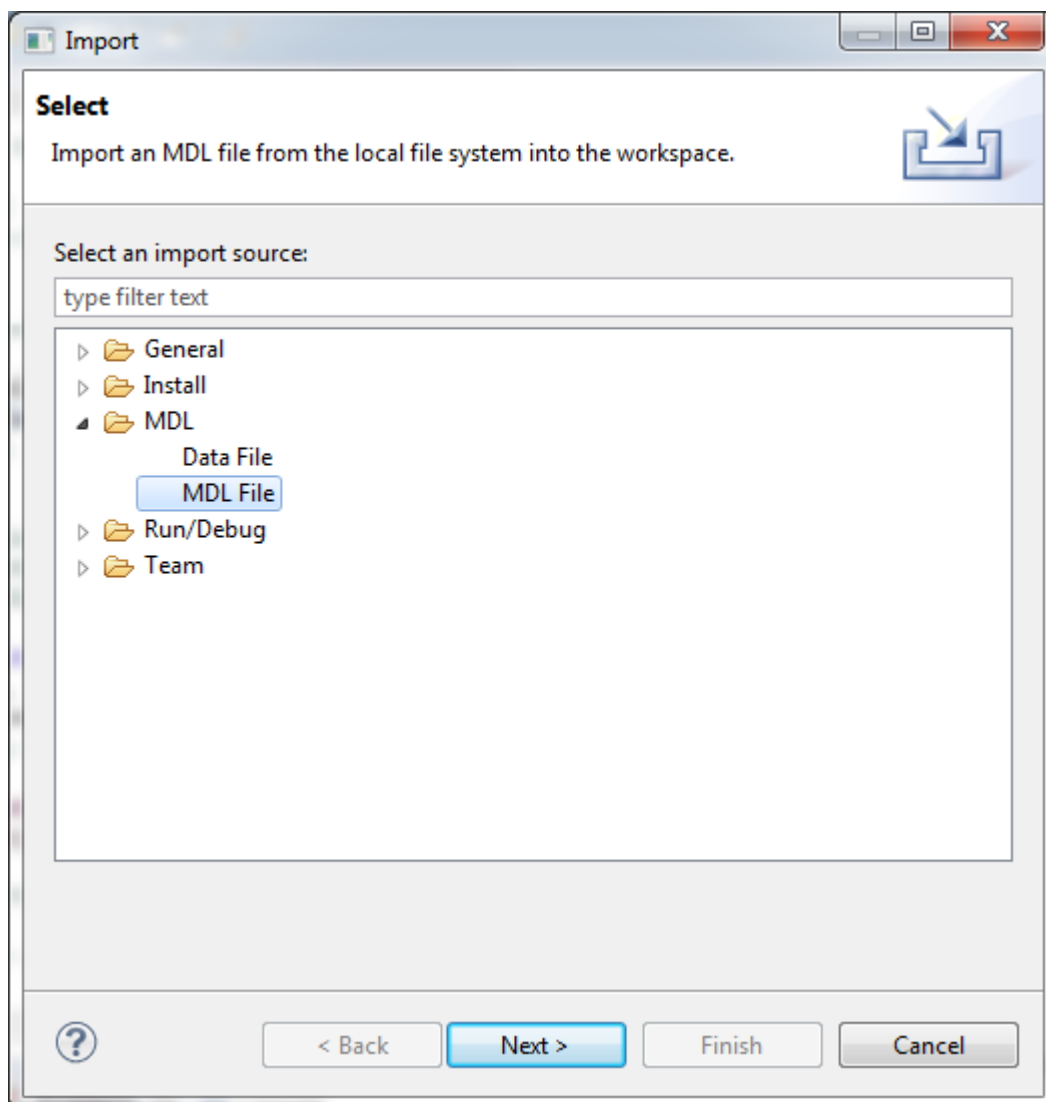
3. In the New wizard, select **MDL Project > MDL File** then click Next.
4. Enter the name for the new file, at this point you can also change the location in which the file will be created.
5. Click **Finish** to create the file.

### 2.6.2 Importing

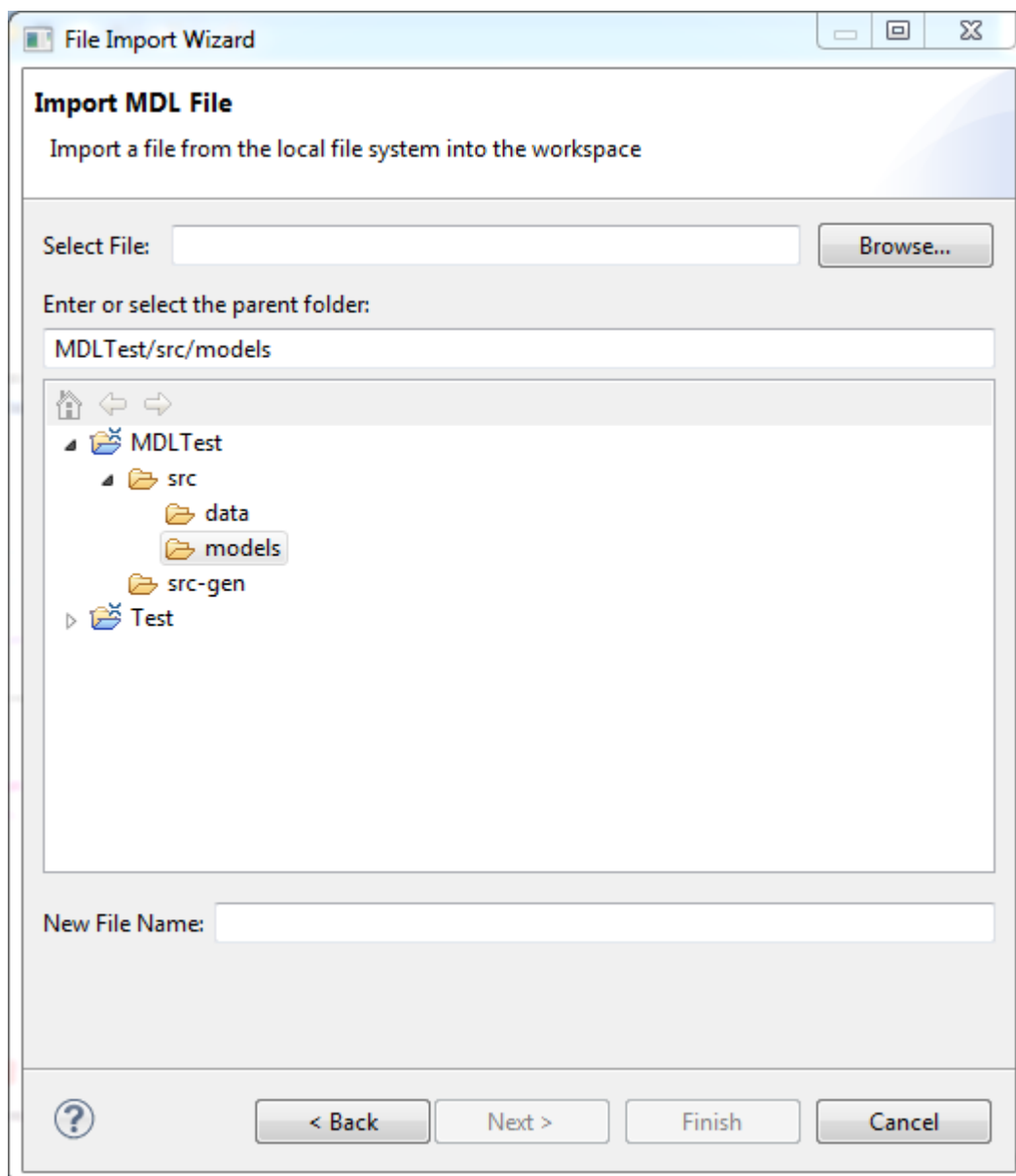
#### 2.6.2.1 MDL

You can import an MDL file into a currently open project.

1. Select the folder to which the file will be added in the Project Explorer.
2. From the menu bar, select **File > Import...**



3. In the Import wizard, select **MDL > MDL File** then click Next.



4. Select the file to be imported by clicking **Browse**. You can change the name under which the file will be imported; you can also change the location in which the file will be created.
5. Click **Finish** to import the file.

### 2.6.2.2 Data

To import data please follow the MDL import procedure (2.6.2.1) substituting point 3 with the following:

3. In the Import wizard, select **MDL > Data File** then click Next.



### 2.6.3 Structure

The MDL Editor allows you to write MDL which conforms to the MCL Specification version 5.0.7.

### 2.6.4 Errors and Warnings

The MDL IDE performs verification on the MDL files in the open projects in the workspace to ensure that they conform to the MDL Specification. The errors and warnings that are generated by this verification are accessible from two places:

- The Problems console tab (see 2.4.4.1)
- The MDL Editor

While the Problems tab simply lists the errors and warnings the MDL Editor shows them in place in the MDL file, which allows fast identification of the problem areas. You can hover your cursor over the highlighted problem to see a description of the issue.

```

        0.01, PPV_V=0.1
    } # end matrix struct1
    #SIGMA
    RUV_PROP_C=list(value=0.1 ,type="VAR")
    RUV_ADD_C=list(value=0.1 ,type="VAR")
}
}# end VARIABILITY
x } # end of parameter object
x missing EOF at '}'
Press 'F2' for focus
# The parameter and variable names in the model object must be matched by
# Note that variables in the data object and parameters may be defined in
# Model variables are derived from the input data variables and parameters
BO1_md1 = mdlobj{
    
```

Modification of the MDL file will cause the re-verification on the fly, meaning that you can see when the error has been resolved without saving the file. The resolved but not saved error will show 'greyed out'.

```

        0.01, PPV_V=0.1
    } # end matrix struct1
    #SIGMA
    RUV_PROP_C=list(value=0.1 ,type="VAR")
    RUV_ADD_C=list(value=0.1 ,type="VAR")
}# end VARIABILITY
x } # end of parameter object
### Model object
# The model object receives a list of input data variables from the data o
# The parameter and variable names in the model object must be matched by
# Note that variables in the data object and parameters may be defined in
# Model variables are derived from the input data variables and parameters
BO1_md1 = mdlobj{
    
```

Once the file is saved the error will disappear and the Problems tab will be updated.

## 2.6.5 Syntax Highlighting

The MDL Editor will highlight the syntax of the MCL in order to make it easily recognisable which words form part of the language specification.

```

INPUT_VARIABLES{
# A name for each input data variable is required. Special uses of input dat
  ID=list(use=id, level=2)
  TIME=list(use=idv, units="h")
  DOSE=list(type=continuous, use=covariate)
  DV=list(type=continuous, use=dv, level=1)
}# end INPUT_VARIABLES

```

## 2.6.6 Auto Completion

The MDL Editor provides auto completion of code using *Content Assist*. Content assist provides you with a list of possible options when you are writing the MDL. The content assist options that are provided are based on the grammar of the MCL.

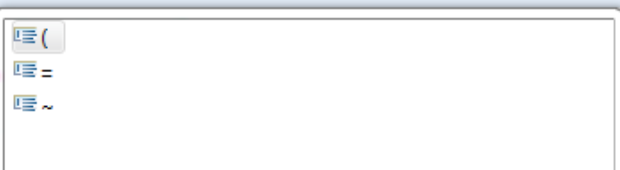
You can access the content assist while typing by pressing Ctrl + Space.

```

RANDOM_VARIABLE_DEFINITION{
# The RANDOM_VARIABLE_DEFINITION block is used to define the distribution of
eta_PPV_CL ~ (type=Normal, mean=0, variance=PPV_CL,level=ID)
eta_PPV_V ~ (type=Normal, mean=0, variance=PPV_V,level=ID)
eps_RUV_PROP_C ~ (type=Normal, mean=0, variance=RUV_PROP_C,level=DV)
eps_RUV_ADD_C
}# end RANDOM_VARIABLE_DEFINITION

INDIVIDUAL_VARIABLE_DEFINITION{
# The INDIVIDUAL_VARIABLE_DEFINITION block is used to define the distribution of
# If this is not a p

```



This shows the options available at this point in the statement; you can select one of these using the cursor keys and press return to have it inserted for you.

The video “MDL IDE Auto Complete” also covers these steps.

## 2.6.7 Templates

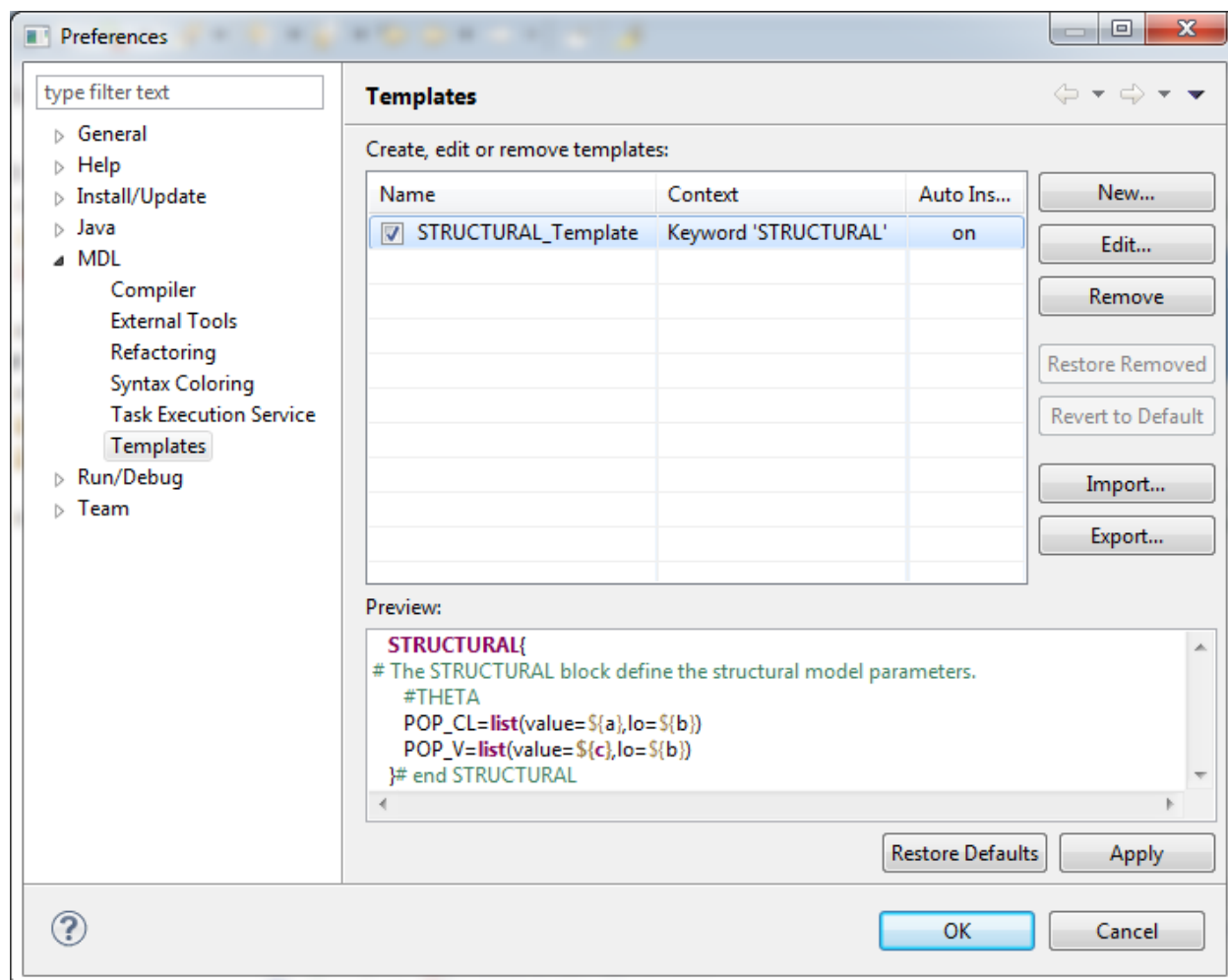
Templates are a powerful feature of the MDL Editor enabling you to pre-define ‘snippets’ of MDL that you can reuse, and in addition they can contain variables that are substituted when the template is applied. This means that you can have a snippet that you can insert and customise with ease.

The video “MDL IDE Templates” also covers these steps.

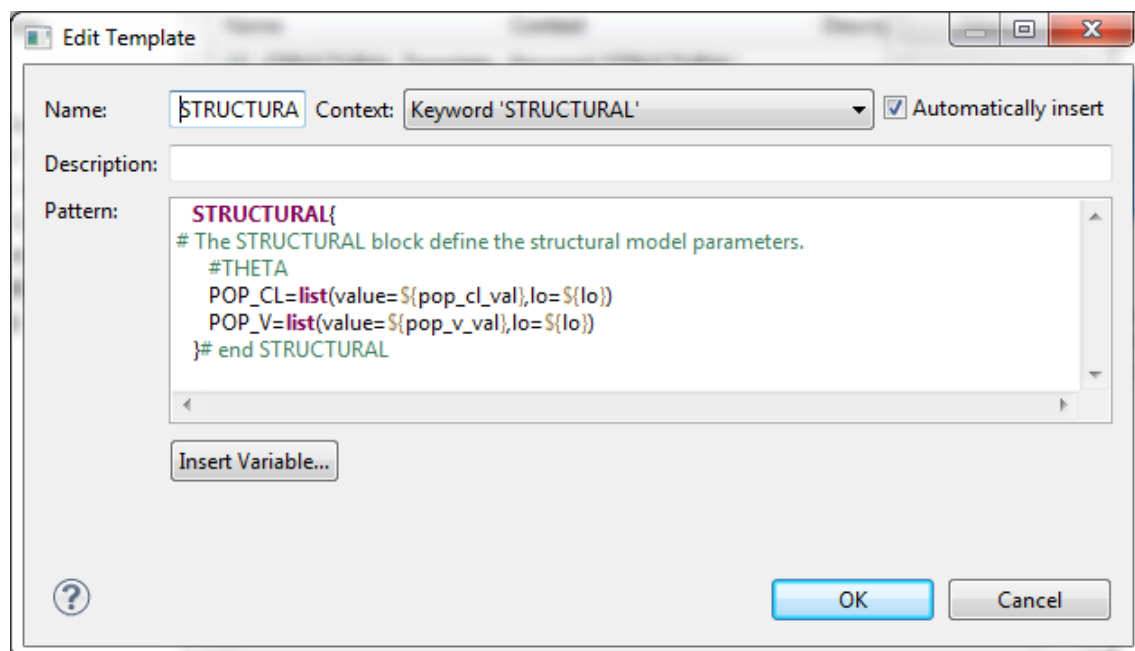
### 2.6.7.1 Creating

You can create templates using the MDL Preferences page.

1. From the menu bar, select **Window > Preferences**



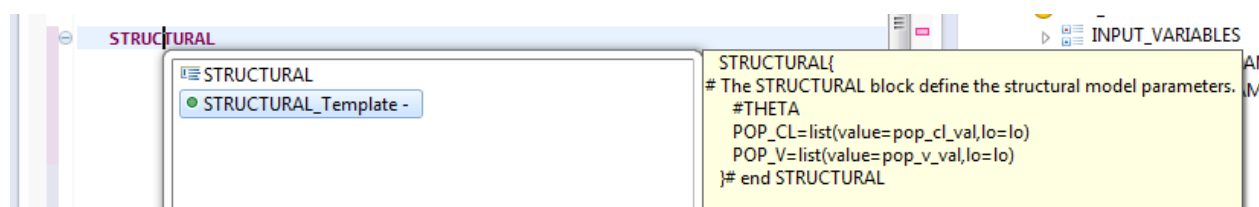
2. In the Preferences, select **MDL > Templates**.
3. Here you can create new templates using the **New...** button as well as editing existing templates with the **Edit...** button.



4. Enter a name for the template, this will be shown when inserting so should be descriptive.
5. Select the context of the template; this determines when the template will be displayed.
6. Enter the MDL you want to be able to insert in the pattern section
7. Enter any values that will change as variables using the **Insert Variable...** button.
  - a. Variables are defined as `${variable_name}`
  - b. In the example above there are three variables with one variable being referenced twice.

### 2.6.7.2 Using

You can insert templates using the Content Assist feature, to activate this feature press Ctrl + Space when in the context of the template.



In the content assist dropdown the template name is shown with a green icon, moving to the template using the cursor keys will also display the contents of the template in a popup. Press return to insert the template.

```

STRUCTURAL{
  # The STRUCTURAL block define the structural model parameters.
  #THETA
  POP_CL=list(value=pop_c1_val,lo=lo)
  POP_V=list(value=pop_v_val,lo=lo)
}# end STRUCTURAL

```

The template has been inserted with the variables highlighted, to replace these with values simply type the value pressing **Tab** to move to the next variable. If a variable is referenced in more than one place then editing it in this way will automatically update all references.

## 2.7 Conversion

### 2.7.1 MDL to NM-TRAN

Currently when a valid MDL file is saved it is also converted into an NM-TRAN format and saved in a NONMEM control file (.ctl). The control file is generated in the src-gen folder of the project and is named the same as the MDL file.

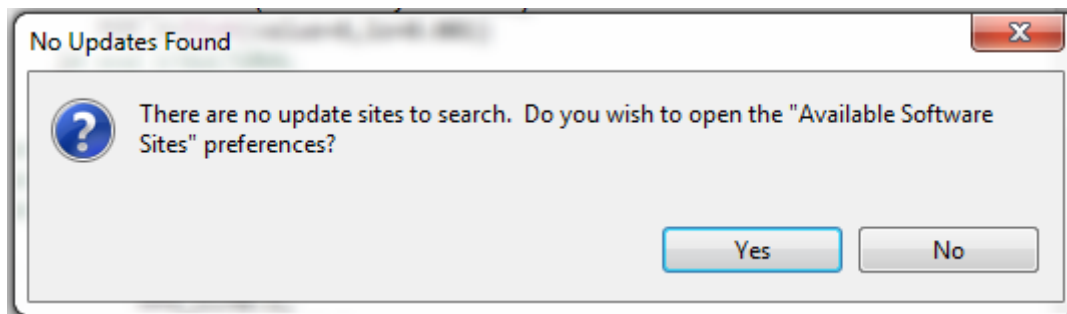
The video “MDL IDE Conversion from MDL to NM-TRAN” also covers this generation.

## 2.8 Troubleshooting

### 2.8.1 The “Run with NONMEM” option fails

This is a limitation of the Public Release as it does not contain the Task Execution Service.

### 2.8.2 The Check for Updates says no update sites to search



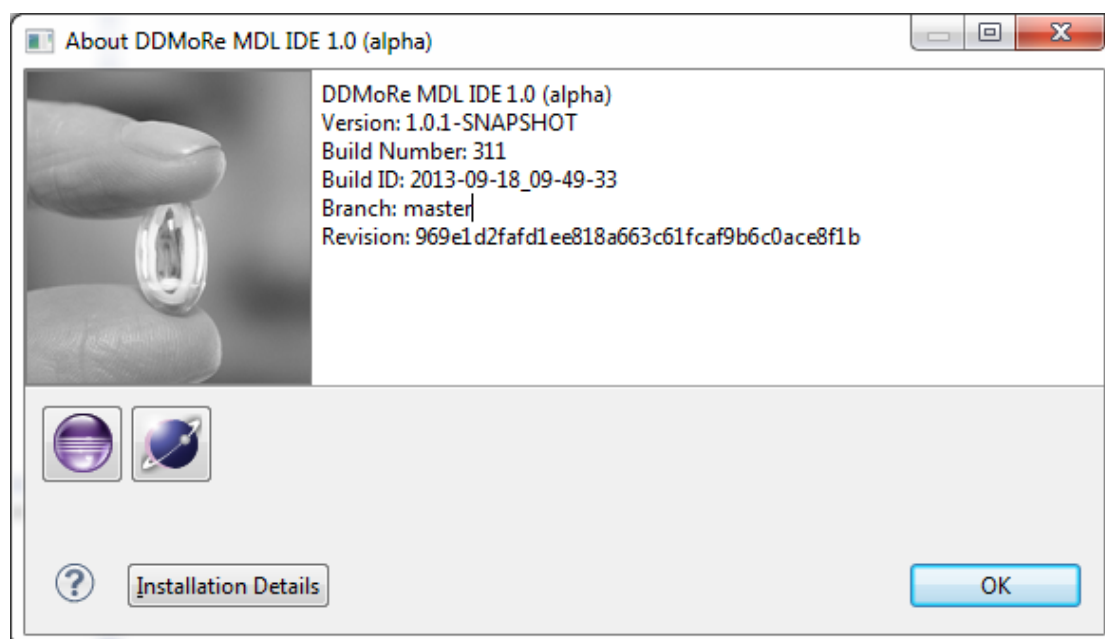
This is a limitation of the Public Release as it does not support the update mechanism.

## 2.9 Additional Help

The MDL IDE is created and supported by Leiden University and Mango Solutions.

For support and to log issues on the MDL IDE please visit the project site hosted on SourceForge: <http://sourceforge.net/p/ddmore/tickets/>

When reporting issues it is very helpful to know the version of the MDL IDE in use. To find this detail select **Help > About DDMoRe MDL IDE** from the menu bar and note the information displayed in the popup.



The eclipse help documentation is a useful resource to find out more about the general capabilities of the IDE (no MDL).

<http://help.eclipse.org/juno/index.jsp>

### 3 Document Revision History

Version	Author	Date	Description
0.1	J Carr	10/09/13	Initial Version
0.2	J Carr	18/09/13	Removed updates section
0.3	J Carr	20/09/13	Added section on how to find IDE version
0.4	J Carr	26/09/13	Added section on importing use cases