ModelManager

Version 1.1

Cherwell Scientific Limited
The Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
United Kingdom

Contents

1. Introduction to ModelManager.....1

About ModelManager.....1
How ModelManager works.....1
System requirements.....2
Installing ModelManager.....2
Activating ModelManager.....3
Technical Support.....4
Contact Cherwell Scientific.....4
About this manual.....5

2. Getting started with ModelManager.....7

ModelManager Basics.....7
Tutorial 1: A quick tour.....8
Tutorial 2: Creating a new study.....11

3. Generating Reports.....21

Creating a report during an analysis.....21
Creating a report from an existing study.....21

4. Study Files.....23

Open an Existing Study.....23

5. ModelManager screens in detail.....27

Study Type Screen.....27
Study Details Screen.....28
Experimental Data Entry Screen.....29
Graph Screen.....31
Analysis Configuration Screen.....33
Configure Parameters Screen.....35
Run Configuration Screen.....38
Optimization Configuration.....40

ModelXchange 42
Settings Dialog 42

6. Constructing Reports 45

Introduction 45
Pre-Requisites 45
Report Worksheets 45
The “General” Sheet 46
The “Filenames” Sheet 51
Other Sheets in the Workbook 51
Final Report Format 51
Format of Presentation Sheets 56
Bitmaps 58
Visual Basic Macros 59
Pseudocode of Report Generation Macros 61
Example: ImportGeneral(…) 63
Example: Adding Records to an Existing Macro 65
Example: Accessing New Tables 67
Worksheets 70
Checklist 71

7. ModelManager Wizard 73

Starting the Wizard 73
Opening Screen 74
Adding Reports 76
Adding Models 77
Wizard Final Screen 90
ModelXchange 94

8. ModelMaker Models 99

Initial Parameter Estimation 99
DT Calculation 100
Events 104

Variable Names 105
Weighting Models 107
Password Protection 108
Model Bitmaps 109

9. ModelManager Databases 111

Introduction 111
Password Protection 112
Clean Study.mdb 118
Config.mdb..... 136

10. ModelManager Script functions 143

Introduction 143
Database Functions..... 143
Calculation Engine 154

Index.....179

1. Introduction to ModelManager

About ModelManager

ModelManager is an application that allows simple and direct control of data analysis for kinetic systems. The user chooses the type of system, the type of mathematical model or models to which the data will be fitted, selects parameters for optimization and runs the analysis. A report is generated in Microsoft Excel. Systems administrators may add new model and analysis types.

The software is essentially a front end application for the popular ModelMaker numerical modeling package. The advantages of ModelManager are:

- The system is designed for use by technicians and operatives who do not need to know about modeling or programming
- The system guides the end user through a series of set procedures
- Data entry is simple
- Data may be previewed in a simple graph display
- New models may be added to extend the system
- Changes to procedures are controlled by systems administrators and cannot be overridden by individual users
- Reports are generated quickly and easily using Microsoft Excel

How ModelManager works

On opening the application the user can choose to start a new study or to view and edit a previously saved study. On choosing a new study the user then selects the type of kinetic system they are modeling. These include in the current release:

- Parent only

- Parent with multiple applications
- Parent and metabolite
- Sediment partitioning
- Parent with two metabolites

The system allows administrators to change or to add to the list of models. The user then adds details of the study - identification, concentration and time units, the date and any useful comments. Data may then be added and the study configured - which models will be fit to the data, whether the data is to be weighted and so on. Finally the analysis run is started.

Models are or have been created using ModelMaker and it is the underlying ModelMaker calculation engine which performs the optimization, i.e. fits the data to selected models and calculates optimum values for the parameters. The end user does not see this application although a full report, including optimized parameters is generated using Microsoft Excel.

System requirements

ModelManager is a 32 bit application therefore requires Windows 95, 98 or NT. The installation requires about 20 Mb disk space and a minimum of 16MB RAM. We do, however recommend at least 32MB of RAM. Reports are produced using Microsoft Excel 95, 97 or 2000 and therefore this application must also be installed on your system. The application will run on a computer with Pentium 75MHz processor or equivalent although P166 should be regarded as a good working minimum.

Installing ModelManager

It is recommended that you quit any running applications while the installation is performed.

- Insert the CD into the CD-ROM drive of your PC.

- Start Windows Explorer and choose the drive corresponding to the CD-ROM

- Locate the executable (.exe) file **ModelManager Setup.exe**

This is the main install file for ModelManager. It will install the ModelManager executable files, all the model files and any reports files defined for your installation.

- Double click the executable file and follow the on-screen instructions

Un-installing ModelManager

To uninstall ModelManager, the user should select "Remove ModelManager" from the ModelManager program group on the Start Menu. The user will be asked for confirmation, then all ModelManager files and directories will be removed from the disk.

Activating ModelManager

The security used by ModelManager locks the software to your computer. When you first install the application you may run it as a demo version for up to thirty days. You can see how much longer you have left by checking the About… box. This is also the place to activate the application as a fully licensed version:

- Select About… from the Help menu

- Once you have opened this dialog click Licensing…

The licensing dialog includes two automatically generated codes. You need to tell Cherwell Scientific both of these codes in order to activate the license. There are two ways to do this.

- Call us directly while you are sitting at your computer

- Tell us both codes

- We will tell you the activation code which you should type into the edit box

- Click OK

Alternatively if it is inconvenient for you to call us directly:

- When you have opened the Registration dialog Click Freeze
- Note the two codes and call or e-mail us at your convenience

If you do not freeze the codes then the next time you open the Registration dialog at ;east one of the codes will be different. You can obtain activation / license code by e-mailing us at:

license@modelmanager.com

or by calling your local Cherwell Scientific office.

Technical Support

If you encounter problems using ModelManager then you will find hints and tips and FAQs (Frequently Asked Questions) on the ModelManager web site:

http: / / www.modelmanager.com

You can also obtain technical support by calling or faxing your local Cherwell office.

Contact Cherwell Scientific

Head Office

Cherwell Scientific Ltd
The Magdalen Centre
Oxford Science Park
Oxford OX4 4GA
UK

Tel: +44 (0)1865 784800
Fax: +44 (0)1865 784801

USA, Canada, Central and South America

Cherwell Scientific Publishing, Inc.
28364 So. Western Ave., Suite 491
Rancho Palos Verdes, CA 90275
USA

Tel / Fax: +1 888-257-6652

Germany, Austria, Switzerland and Eastern Europe

Cherwell Scientific Ltd
c/o CHEM Research GmbH
Hamburger Allee 26-28
60486 Frankfurt
Germany

Tel: +49 (0)69 970841-11
Fax: +49 (0)69 970841-41

e-mail addresses

General support issues: support@modelmanager.com

Registering/licensing ModelManager: license@modelmanager.com

Web site

The ModelManager web site is at:
http://www.modelmanager.com

About this manual

Chapters 2 to 5 of this manual present an end users view of using ModelManager. The remaining chapters present information for systems administrators. The software is distributed with the end users section in Microsoft Word format which may be printed out for individual users. Please note that this is copyright of Cherwell Scientific Limited and cannot be altered. You may append or add new sections that are relevant to models you create and add to the ModelManager application.

2. Getting started with ModelManager

ModelManager Basics

Starting ModelManager

Start ModelManager in any of the following ways:

- Double click the shortcut icon that appears on the desktop
- Select the ModelManager program via the Start | Programs menu on the Windows task bar
- Double click Modelmanager.exe executable file in Windows Explorer

Keystrokes

Navigating the application is mostly accomplished using your mouse, clicking on buttons, drop down lists and check boxes. In addition you type text into edit boxes. However many of the text entries and buttons in the ModelManager windows have keyboard shortcuts associated with them. The appropriate shortcut character for a particular button or text field is underlined in its title. For example, the "Next screen" button on any ModelManager screen is shown below:

The shortcut for this button is therefore the character "n". This is activated by holding down the "Alt" button and then pressing and releasing "n" (read Alt+N). Note that although we use N (upper case) in the manual, it is not case sensitive in the application.

Exiting ModelManager

The ModelManager session is terminated by selecting **Exit ModelManager** from the **File** menu, pressing the Exit button on the opening screen or by clicking the button  on the title bar. Confirmation will be requested before ModelManager closes down, if a study is currently in progress then the user will be prompted as to whether to keep or discard unsaved changes.

Tutorial 1: A quick tour

The ModelManager installation includes the following sample files. Use these to explore the various ModelManager screens.

- PM.sty (Parent-metabolite)
- PMA.sty (Parent with multiple applications)
- PMM.sty (Parent with two metabolites)
- PO.sty (parent only)
- SP Demo.sty (Sediment partitioning)

In this short guided tour we will take a look at the Parent only study - PO.sty.

Step 1: Open an existing study

Start ModelManager from the shortcut icon on your Windows desktop or the Start|Programs menu. You are presented with the opening screen.

- Click 2. Open an existing study
- In the Open dialog select the file PO.sty and click Open

A message warns you that the ModelManager database is being accessed and then you are presented with the Study details screen. This lists important information about the study - identification, original date, concentration and time units. To continue either:

- Click Next >>
- Select Next screen from the Screens menu

- Click the Next screen icon on the Shortcuts toolbar

Notice that the screens menu lists all the available screens in ModelManager but most are grayed out. Only the Experimental data entry screen is available as this is indeed the next screen in the procedure. As you progress you will find that you can step back to modify any of the previous stages. Before progressing to the next

screen check out the Measurement units and Time units drop down lists. These contain a number of selections which can be edited by the systems administrator or supervisor. Also note the Number of datasets edit box - this has consequences for the next screen.

Step 2: The Experimental data entry screen

This screen comprises a number of tabbed panels - one for each dataset. This was actually defined in the Number of datasets edit box in the Study details screen. To step between datasets:

- Click on a tab to open the panel and display the data

The panel includes some important information about the dataset, comments and, of course, the data itself. To edit the data:

- Click in a cell in the Experimental data table
- Type a new value
- Press Enter or use the arrow keys to navigate to a new cell

To preview the data displayed:

- Click the button labeled Graph

This opens the Graph screen. This is a basic graph display but gives you a good idea about the status of the data in the study. Note that you can select a logarithm display and display points rather than a line graph. You cannot print this graph as it is only intended as a preview. Click Close to return to the data entry screen.

- To continue the analysis Click Next>>

Step 3: Configure the analysis

The Configure Analysis screen comprises several panels. In the top left corner you can choose any combination or all of the datasets that have been created:

- Click on the name to select
- Click on a highlighted (selected) name to deselect

In the top right hand corner a choice of three radio buttons enables selection of model parameters. These are described in detail in the next tutorial but briefly these are defined as:

- Use existing parameter estimates

  If the analysis is being run for the first time, then these will be the default parameter values for each model. If analysis has already been performed for a particular model / dataset pair, then the results of the previous analysis will be used

- Use the user inputs to calculate initial estimates

  If you have entered estimates for DT in the previous screen then checking this selection will use these for the calculation.

- Perform parameter search to find initial estimates

  ModelManager uses its own methods to estimate parameter values.

At the bottom left you may select which models are to be run during the analysis (Models to fit) and whether the data is to be weighted. Report DTX at: allows you to specify the value at which ModelManager is to calculate the Degradation Time. In addition to a calculation at this specified time, values for DT50 and DT90 are calculated by default, with errors corresponding to 95% confidence via the appropriate Student t value.

Configure parameters opens the Configure Parameters screen and allows you further control of the parameters to be used. Take a quick look at this screen and then click Cancel to proceed with this trial analysis.

Step 4: Run the analysis

Still in the Configure Analysis screen:

- Click Go

The analysis run starts and the Calculation Status dialog maintains a log of the process. When the analysis is complete Excel starts

automatically. You will be warned that the Excel spreadsheet contains macros and should click Yes to accept and open the report.

Step 5: View the results

Switch to Microsoft Excel to view the report. The report lists study details and plots fitted results. A different worksheet is created for each model chosen and another compares the fitted data. Once created this can be treated like any other spreadsheet - so you can print, copy and paste selections and import into other applications such as your word processor. Note that you can set the location where the report is stored using the Settings dialog opened from the Edit menu.

Step 6: View results of an existing study

You can choose to view the results of an existing study from the opening screen of ModelManager. If you do this then you are prompted to open a file and then presented with the View Existing Study screen. This lists details about the study and allows you to generate a report in Excel.

And finally: The Shortcut Bar

To the top of the main screen, there is a shortcut toolbar which allows easy access to various screens. Use this to move quickly between screens, create, open and save studies in ModelManager. You will notice that when certain screens or functions are unavailable then the relevant toolbar buttons are grayed out.

The Shortcut toolbar

On each screen, various buttons on this toolbar are enabled and disabled, as different options are available on each screen.

Tutorial 2: Creating a new study

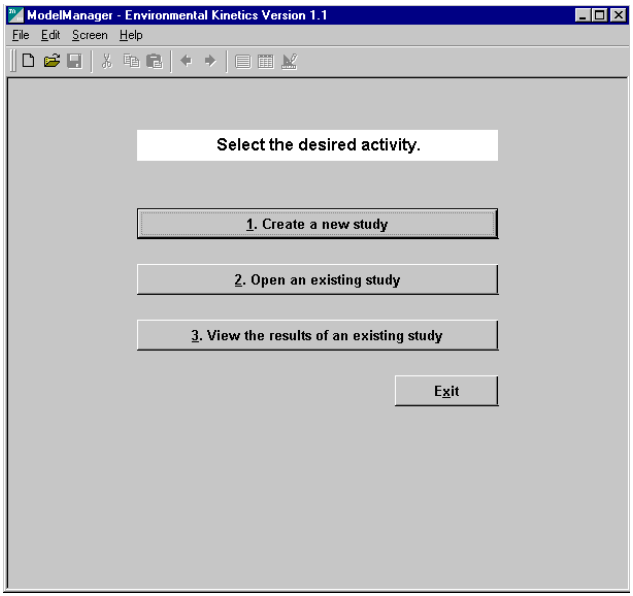
The section will describe how a study can be created, from inputting the data through to generating a report. We also include some more detailed information about the various screens.

Step 1: Create a new study

From the opening screen:

- Click 1. create a new study

The opening screen of ModelManager.



Step 2: Select the study type

Once you have decided to create a new study, you need to specify the type of study which he wants to create. This is done with the study type screen. There are five study types available in ModelManager and the screen has one button for each screen.

The Study Type screen contains five buttons according to the study types available.

Parent Only

This study is used when there is one data series which needs to be analyzed, for example when there is one compound that degrades over time

Parent with Multiple Applications

As with the previous study type, there is only one compound that degrades over time. However, extra quantities of the compound can be introduced into the system as time passes.

Parent and Metabolite

In this study type, as the parent compound degrades another chemical is produced. This study is used to track how they both degrade over time.

Sediment Partitioning

This study type is used to examine how a chemical is absorbed by the sediment in a system from a water column, and how they subsequently degrade over time.

Parent with two metabolites

This study is identical to the Parent and Metabolite case except that two chemicals are formed in the system.

Once a study type has been selected, it may not be altered. If you wish to change the study type, you must quit the current study and begin a new study of a different type.

When the button is pressed, an empty study of the type selected is then created and the user is presented with the Study Details screen. For the purposes of this example, we will create a Parent Only study, so press the Parent Only button.

- In the Select desired study type screen click 1. Parent only

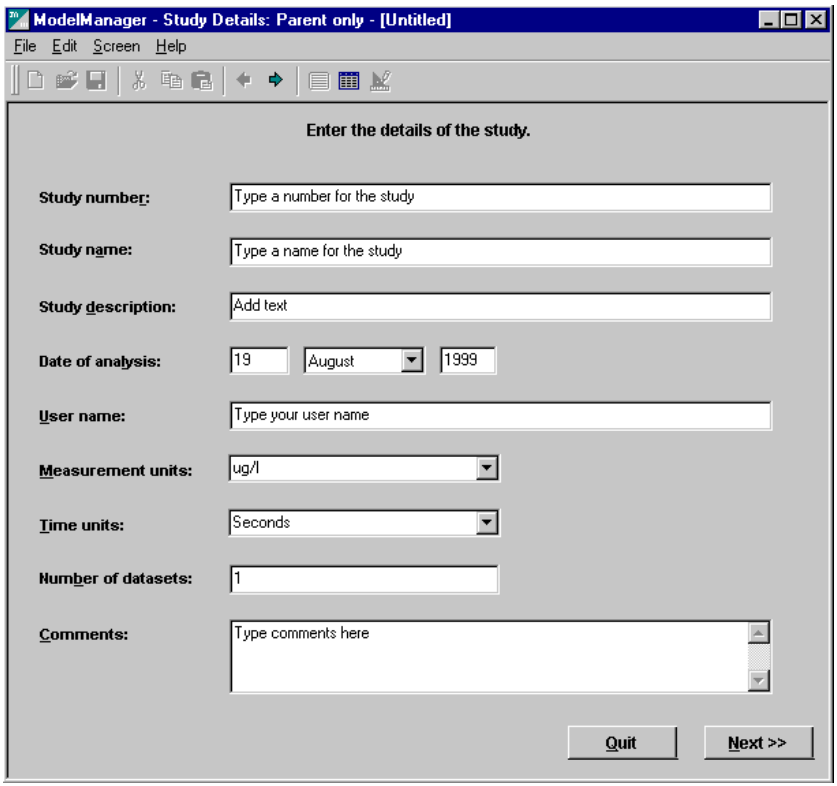
The application pauses (you will see the egg timer icon) while the database is opened and a new study created.

Step 3: Add study details

Once a new study or an existing study has been opened, the user is presented with the **Study Details** screen. This is shown below. The study details screen enables the user to enter comments and specific information about the study. This information can then be shown on the generated report.

You can enter any information you wish here as it does not affect how the analysis performs. In this example we shall enter two sets of data, so press “Alt-b” to go to the number of datasets field and type ‘

The Study Details screen



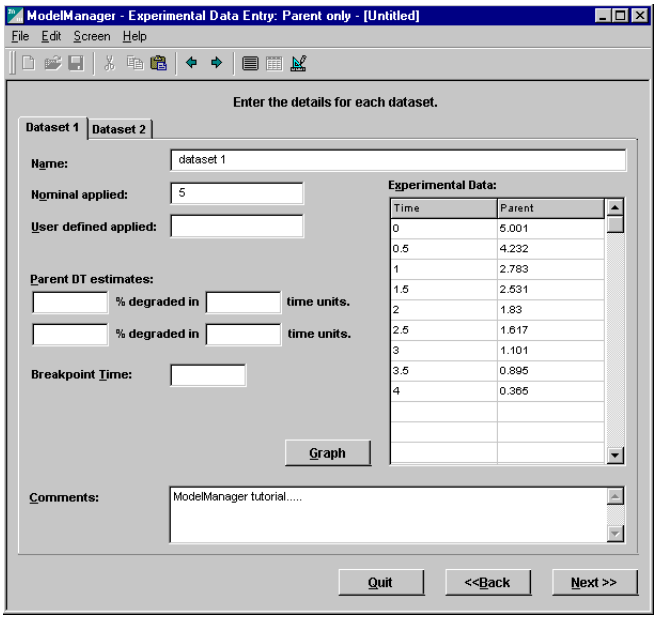
- Type relevant text in the Study number, Study name, Study description and User name edit boxes
- Select measurement and time units from these drop down menus[1]
- Type **2** in the Number of datasets edit box to allow us to add two sets of data
- Click Next>> to proceed to the Experimental data entry screen

[1] note that you can actually change these for this study by typing text in the edit box though they will not be added to the drop down list unless your systems administrator or ModelManager supervisor changes the database directly.

Step 4: Add experimental data (1)

The screen for experimental data entry in the parent only study is shown below. This screen is used to record the experimental data measured during a trial run.

The Experimental Data Entry screen



In this example we will use two sets of data. First fill out the rest of the screen:

- Press <Alt+A> to select or click in the Name edit box and give the first set of data the name **Dataset 1**

In this experiment we know that we put 5 units of the chemical into the system:

- Tab to or click in the Nominal Applied edit box and enter **5**

It is not necessary to enter any estimates of how the compound degrades To add the data

- Select the data grid and enter the following

time	Parent
0	5.001

0.5	4.232
1	2.783
1.5	2.531
2	1.83
2.5	1.617
3	1.101
3.5	0.895
4	0.365

Note that if you have data in digital form - even a simple text file, you can copy and paste the data directly into the table in this screen.

Step 5: Add experimental data (2)

Now enter the data for the second set of data.

- Clicking the Dataset 2 tab at the top of the screen.
- Give this the name **Dataset 2**
- Set the nominal applied to be 500.
- Enter the following data below

Experimental Dataset 2

time	Parent	time	Parent
0	503	100	139
10	424	110	128
20	370	120	125
30	298	130	115
40	262	140	111
50	213	150	104
60	192	160	105
70	155	170	98
80	140	180	88
90	135		

It is possible to get a rough idea of how the compound has degraded by clicking the “Graph” button (Alt+G). The data shown on the graph corresponds with the selected dataset tab on the data entry screen.

From this you can see that the data in dataset 1 degrades with roughly a straight line, and in dataset 2 it follows a curve –although the curve becomes noticeably shallower after time 80.

- Press the Next>> button on the data entry screen to move to the final step in the study, the analysis configuration screen

Step 6: Analysis Configuration, run 1

As the name suggests, is where the analysis run is configured. The screen is linked through various options to several other screens. However for this tutorial we will first analyze Dataset 1

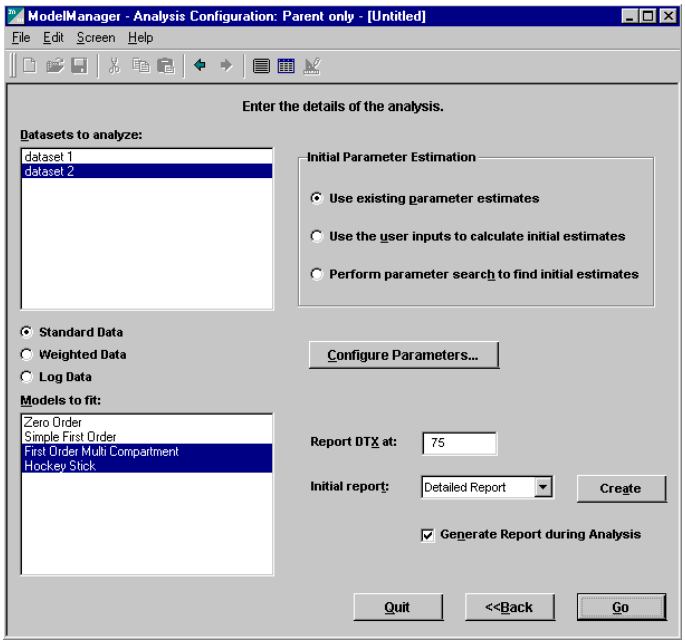
- Click to select Dataset 1
- Check Standard data
- Select Zero Order Simple First Order
- Click Go

Step 7: View the result, run 1

- Switch to Excel to view the report

Note that three sheets are generated, one for each model and one which compares the fits (both in output data and as a plot)

The Analysis Configuration screen



Step 8: Analysis configuration, run 2

Switch back to ModelManager and you should still have the Analysis Configuration screen in view. To perform an analysis run for a different dataset (note that you can in general select more than one dataset):

- Deselect Dataset 1 and select Dataset 2
- Deselect Zero Order and Simple First Order and select First Order Muli Compartment and Hockey Stick
- Click Go

Step 9: View the results

Once again switch to Excel and view the report. You will find sheets for both models and a third showing a comparison.

The elements of the Analysis Configuration screen are as follows:

Datasets to Analyze

The user should select those datasets which are to be included in the analysis from this list. Several datasets may be selected by clicking on their list item. Selected datasets are de-selected by clicking their list item again.

Models to Fit

The user may select which models are to be run during the analysis by selecting the model names from this list. These models will be applied to all datasets selected from the dataset list.

Weight the experimental data check box

This box should be checked if the experimental data is to be weighted. This causes ModelManager to select the appropriate weighted models.

Initial Parameter Estimation

This is a series of options for the initial estimation of the model parameters. Only one option may be selected from this list. There are three options available for all study types except for Sediment Partitioning, for which there are two. The options are:

Use existing parameter estimates

This option takes the parameter estimates which are contained in the database for all models selected for analysis. If the analysis is being run for the first time, then these will be the default parameter values for each model. If analysis has already been performed for a particular model / dataset pair, then the results of the previous analysis will be used. If parameter values have been amended using the "Configure Parameter" screen, the amended values are taken.

Use the user inputs to calculate initial estimates

With this option selected, the initial parameter estimates are taken from user inputs on the Experimental Data Entry, namely the estimates of the percentage of parent (and metabolite) degraded in specified time. In the case of a Parent Only study type, the "Hockey del also uses the "breakpoint time" on this screen. If the entered degradation estimates are incomplete, ModelManager estimates values from the experimental data. This option is not available for Sediment Partitioning studies.

Perform parameter search to find initial estimates

This option uses the grid search or simulated annealing method to calculate initial values for the parameters. ModelManager first calculates an estimate for each parameter from the user inputs as above which is then used as the basis of the parameter search.

Configure Parameters button

The button named “Configure Parameters…” opens a screen via which the default parameters may be altered for the currently selected models and datasets.

Run Configuration menu item

This takes the user to the Run Configuration screen, on which the default run options may be overwritten. Access to this screen is limited to those with Super-user authority. The Run Configuration screen is described in Section 0.

Optimization Configuration menu item

This takes the user to the Optimization Configuration screen, on which the default optimization options may be altered. Access to this screen is limited to those with Super-user authority.

“Report DTX at” box

This box allows the user to specify the value at which ModelManager is to calculate the Degradation Time. In addition to a calculation at this specified time, values for DT50 and DT90 are calculated by default, with errors corresponding to 95% confidence via the appropriate Student t value.

The report (which should be automatically opened if requested) at the end of the analysis should be specified under the heading “Initial report”. The list of available reports may be altered by means of the ModelManager Wizard tool.

If a report is not desired during the current analysis, the check box under the heading “Generate Report during Analysis” should be de-selected. The user may create a report at any time (while an analysis is not currently running) by clicking the “Create” button. This button is not available until an analysis has already been run.

If a report has been requested to be generated automatically at the end of the analysis, and any of the model optimizations have not been successful, then the user is prompted, at the end of the analysis, as to whether they wish to proceed with report generation.

3. Generating Reports

Creating a report during an analysis

To create a report during an analysis ensure that Generate Report during Analysis (Analysis configuration screen is checked.

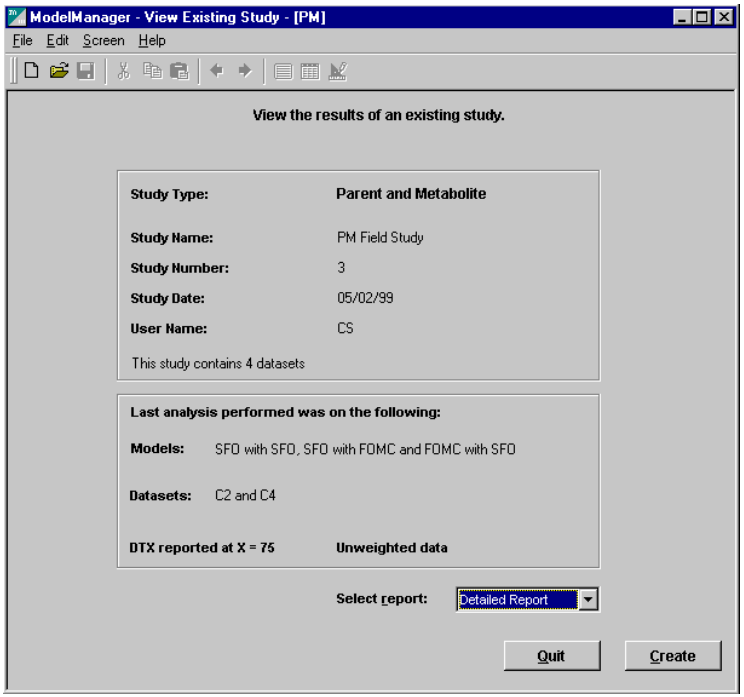
Creating a report from an existing study

If the user has chosen to view the results of an existing study in the opening screen, they will be presented with an Open file type dialog.

- Select a file and click Open

You are then presented with the View existing Study screen.

The View Existing Study screen



This screen provides an alternative way of opening a study. You are presented with a brief summary of the study results, and offered a choice of reports that may be generated. (Select report drop down list). Note that if a report is not currently “allowed”, i.e. no analysis has taken place, or the user has changed the configuration of the analysis since the last analysis was done, then the “Create” button will be grayed out.

The elements of the screen are as follows:

Summary This summary lists some details of the study that was selected. This may assist in verifying the selected study is that required.

Select Report This lists the available reports for a particular study type. The user should select the desired report from this list.

Create button This button should be pressed to create the selected report. Note that if a study has been created, saved and exited without any models having been run, this button will not be available, as there will be nothing to report. In this case the user should open the study and run the analysis before this option is available.

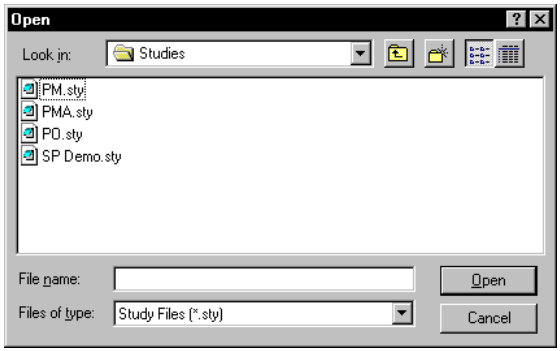
4. Study Files

Open an Existing Study

Once the desired study has been selected, you are automatically presented with the Study Details screen, containing the information of the selected study.

On the opening screen of ModelManager, you have the option to open a previously existing study. The user is presented with a dialog which allows them to select which study they wish to open. This dialog is shown below.

Open Study File dialog box



A study file has the extension .STY and is merely a pointer to a database containing all the information relating to that study. For example, if we have saved a study under the name TEST.STY, we would have the following:

- A study file TEST.STY which contains the name of the study database
- A study database named TEST.MDB, in the same folder as TEST.STY, which contains the study data such as datasets, results etc.

When a study is opened, the study database is copied into the installation directory of ModelManager, which is by default:

C:\Program Files\Cherwell Scientific\ModelManager (EK)

The database is renamed mman.mdb, and it is this copy of the study database that may be altered by the user when the study has been opened.

When the user chooses to close a study that is currently open., they are prompted as to whether the study should be saved. If the user selects “yes” then the database mman.mdb (in the installation directory of ModelManager) is copied back to the original location of the opened study. For example, if we had opened the file:

C:\Program Files\Cherwell Scientific\ModelManager (EK)\Studies\TEST.STY

we would have copied the file:

C:\Program Files\Cherwell Scientific\ModelManager (EK)\Studies\TEST.MDB

to the “working” location:

C:\Program Files\Cherwell Scientific\ModelManager (EK)\mman.mdb

Once the user has chosen to save the study, then this file above is copied back to the original location:

C:\Program Files\Cherwell Scientific\ModelManager (EK)\Studies\TEST.MDB

The original file is overwritten.

If the user closes the opened study without choosing to save the file, their changes are discarded. The file mman.mdb is not copied back to the original study location and the original file TEST.MDB is left unchanged.

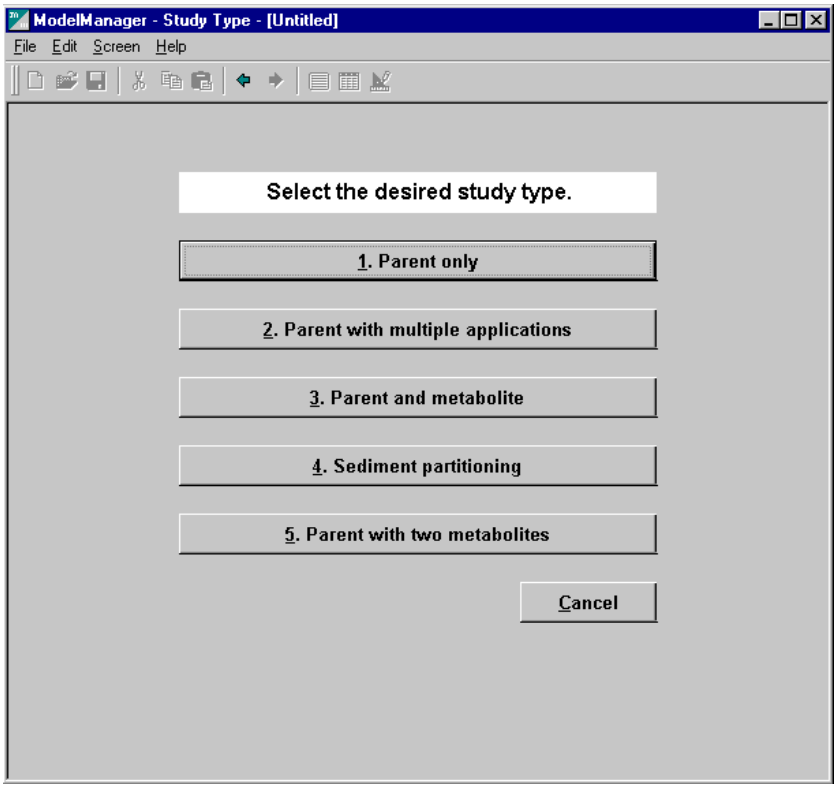
As the study database contains all information as to the status of every control on the ModelManager screens, and details on datasets entered by the user and the results of their analyses, then all of this information is saved when saving a study.

5. ModelManager screens in detail

Study Type Screen

On choosing to create a new study, you are prompted to select a study type by the Study Type screen.

The Study Type screen



Study Type buttons

The buttons describe the study types available. Select one to proceed.

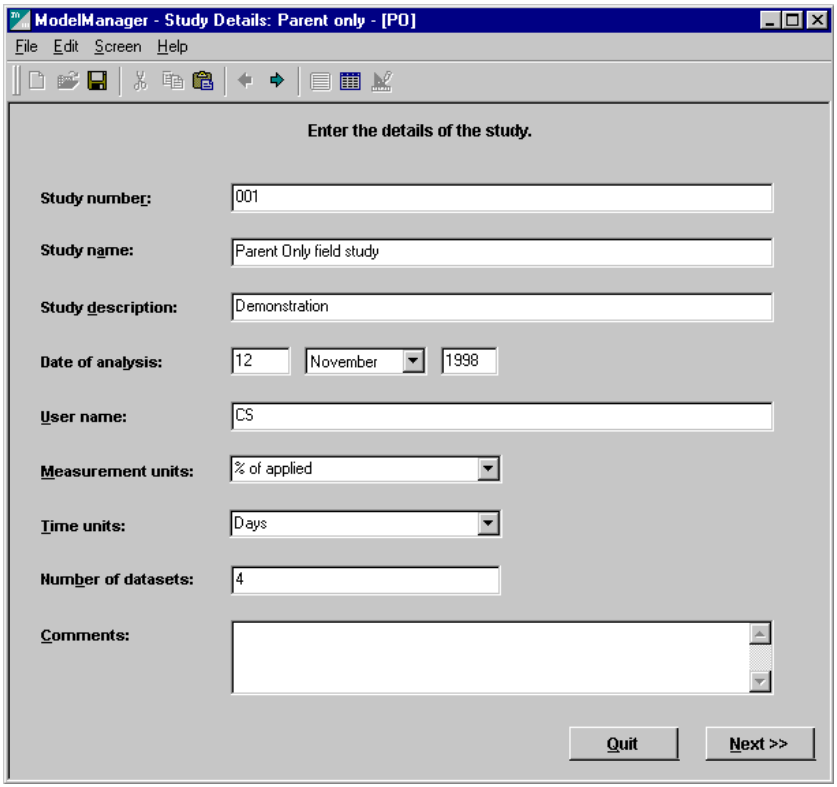
Cancel button

Clicking Cancel quits the current study and takes the user back to the opening screen.

Study Details Screen

Once a new study or an existing study has been opened, the **Study Details** screen is displayed.

The Study Details screen



This allows entry of comments and specific information about the study. All the data entered is available when generating reports.

Study details

Use the Study details edit boxes to specify details which identify the study in question. These are:

- Study number
- Study name
- Study description
- User name

- date of analysis (see below)

Date of analysis

The date field on this screen checks the values entered to ensure they are valid. Leap years and year 2000 are handled correctly. If the user enters an invalid date, they will be prompted for a new entry.

When creating a new study, the date field contains the current date. When opening an existing study, the date field is not amended to show the current date.

Measurement units, Time units

These list the range of Measurement and Time units available in the database. You may enter their own time and measurement units if the unit they require is not available.

Number of datasets

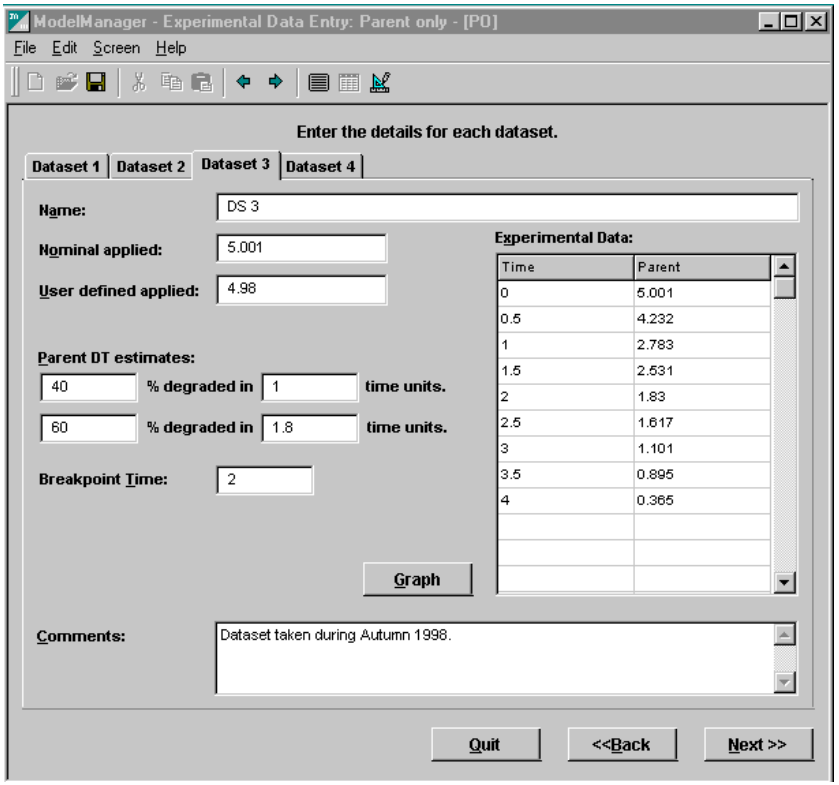
Enter the number of datasets that should be contained in the study. The number of datasets in a study may be adjusted at a later date by returning to the Study Details screen. For all new datasets created, the database is filled with default values for the parameters of all the models relevant to that study type.

There is an upper limit of 10 datasets per study and numbers greater than this limit will be truncated to 10. The default value of 1 (one) is used if zero or negative numbers are used.

Experimental Data Entry Screen

The data entry screen presented to the user by ModelManager will depend upon the study type in question. For the purposes of this description, we shall refer to the screen corresponding to the Parent Only study type.

The Experimental data entry screen



Dataset Tab Controls

All the datasets are available for configuration on a single screen. The Tab control allows switching between datasets (labeled Dataset 1, Dataset 2 etc.).Choose a particular dataset by clicking the appropriate tab.

The available fields on the Experimental Data Entry screen are listed and described below. If a field is not available for a particular study type, this is mentioned in the description of that field.

Dataset name

Enter a name to identify a particular dataset. This name is available for insertion into Reports.

Nominal Applied/User Defined Applied

These relate to the initial amount of parent applied and provide alternative methods of specifying this quantity. Many models contained in ModelManager contain a parameter relating to the initial amount applied, and these extra fields give more flexibility to the user when setting an initial value for this parameter in the optimization routine.

Parent Estimates Enter the value of estimated degradation times for the parent. Note that these figures are optional and may be used on a later screen to calculate the initial parameter values in the optimization. They are, however not available for the Sediment partitioning study type.

Metabolite Estimates For the Parent and Metabolite study type the user enters here the value of estimated degradation times for the metabolite. These figures are also optional and may be used on a later screen to calculate the initial parameter values in the optimization. Note that they are only available for the Parent and Metabolite study type.

Breakpoint Time This field is used only when calculating initial parameter values for the Parent Only Hockey Stick model. It is not required for any other model or study.

Experimental Data This table is used for the entry of numerical experimental data. The time values do not need to be entered in chronological order, as they are re-ordered automatically on leaving the screen. The data entered may be viewed by clicking the **Graph** button.

Comments Any comments the user might like to associate with the dataset.

Graph Button Clicking this button sends the user to the Graph screen. This screen is available to all study types, and is described in Section 0.

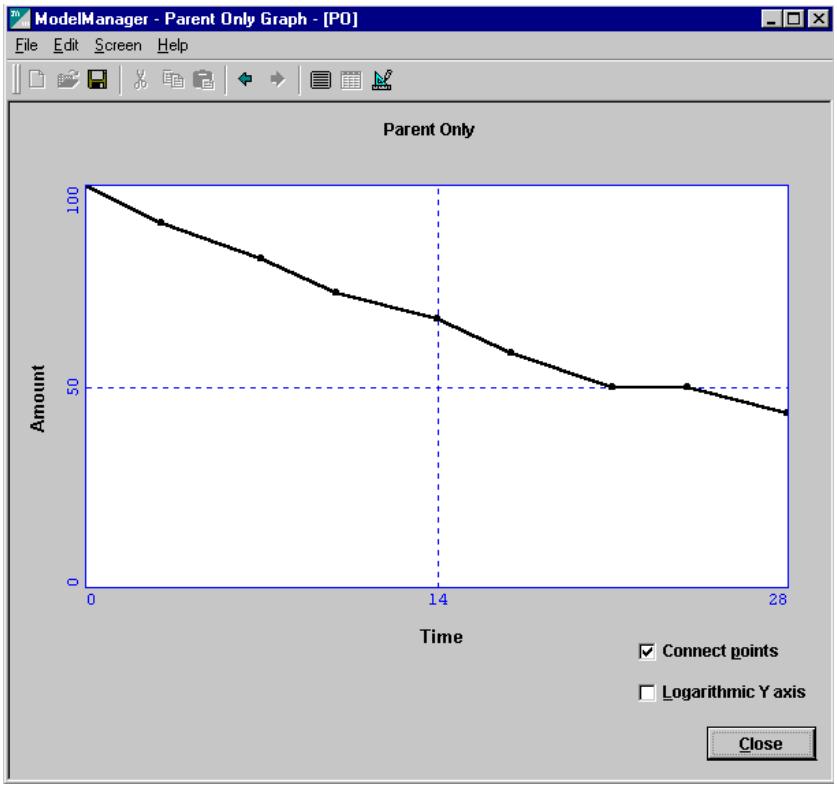
Graph Screen

The appearance of the Graph screen depends on the current study type. For the Parent Only and Multiple Applications study types it contains a single trace, and has two traces for the other two study types.

Graph plot area This displays the data entered in the table of the Experimental Data Entry screen. If no data is available to be plotted, the plot area will not be visible and the user will be shown the information “Empty

Graph Properties There are two properties which may be adjusted by the user. These are:

The Graph screen



- **Disconnect Points / Connect Points**

  The user may choose to display the data with or without a line connecting the points by clicking the button labeled “Disconnect Points”. The line will be removed from the graph and the button will then be labeled “Connect Points”. Clicking again will re-connect the points and so on.

- **Logarithmic Y axis**

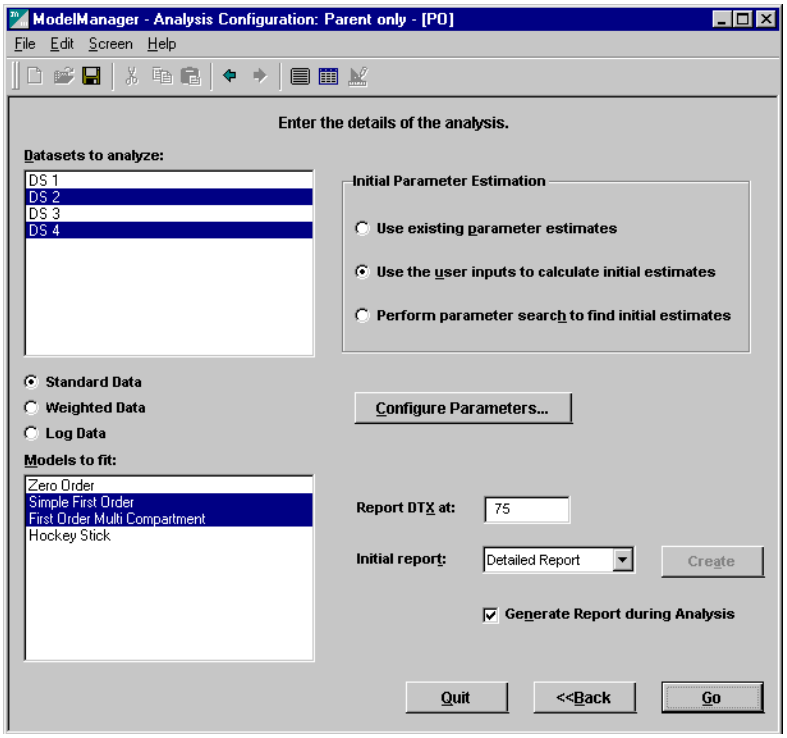
  The user may select the Y axis to have a logarithmic scale. Un-checking the button will cause the Y axis to become linear.

Close button Clicking the Close button takes the user back to the Experimental Data Entry screen.

Analysis Configuration Screen

This is the focal point of the application, where the analysis run is configured. The screen contains links to various option screens.

The Analysis Configuration screen



Datasets to Analyze

The user should select those datasets which are to be included in the analysis from this list. Several datasets may be selected by clicking on their list item. Selected datasets are de-selected by clicking their list item again.

Models to Fit

Select which models are to be run during the analysis by clicking the model names from this list. You may select any combination of the listed models. Click the model again to deselect. The selected models will be applied to all datasets selected from the dataset list.

Weight the experimental data radio buttons

Check one of the three radio buttons to choose whether the data analysis is to be weighted. This causes ModelManager to select the appropriate weighted models.

Initial Parameter Estimation

This is a series of options for the initial estimation of the model parameters. Only one option may be selected from this list. There are three options available for all study types except for Sediment Partitioning, for which there are two. The options are:

- **Use existing parameter estimates**

  This option takes the parameter estimates which are contained in the database for all models selected for analysis. If the analysis is being run for the first time, then these will be the default parameter values for each model. If analysis has already been performed for a particular model/dataset pair, then the results of the previous analysis will be used. If parameter values have been amended using the "Configure Parameter" screen, the amended values are taken.

- **Use the user inputs to calculate initial estimates**

  With this option selected, the initial parameter estimates are taken from user inputs on the Experimental Data Entry screen namely the estimates of the percentage of parent (and metabolite) degraded in specified time. In the case of a Parent Only study type, the "Hockey Stick" model also uses the "breakpoint time" on this screen. If the entered degradation estimates are incomplete, ModelManager estimates values from the experimental data. This option is not available for Sediment Partitioning studies.

- **Perform parameter search to find initial estimates**

  This option uses the grid search or simulated annealing method to calculate initial values for the parameters. ModelManager first calculates an estimate for each parameter from the user inputs as above which is then used as the basis of the parameter search.

Configure Parameters button

The button named "Configure Parameters…" opens a screen via which the default parameters may be altered for the currently selected models and datasets.

Run Configuration menu item

This takes the user to the Run Configuration screen, on which the default run options may be overwritten. Access to this screen is limited to those with Super-user authority.

Optimization Configuration menu item

This takes the user to the Optimization Configuration screen, on which the default optimization options may be altered. Access to this screen is limited to those with Super-user authority. The Optimization Configuration screen is described in Section 0.

"Report DTX at" box

This box allows the user to specify the value at which ModelManager is to calculate the Degradation Time. In addition to a calculation at this specified time, values for DT50 and DT90 are calculated by default, with errors corresponding to 95% confidence via the appropriate Student t value.

Initial Report / Create button / Generate Report checkbox

The report that should be automatically opened (if requested) at the end of the analysis should be specified under the heading "Initial report". The list of available reports may be altered by means of the ModelManager Wizard tool (see the separate documentation on the ModelManager Wizard).

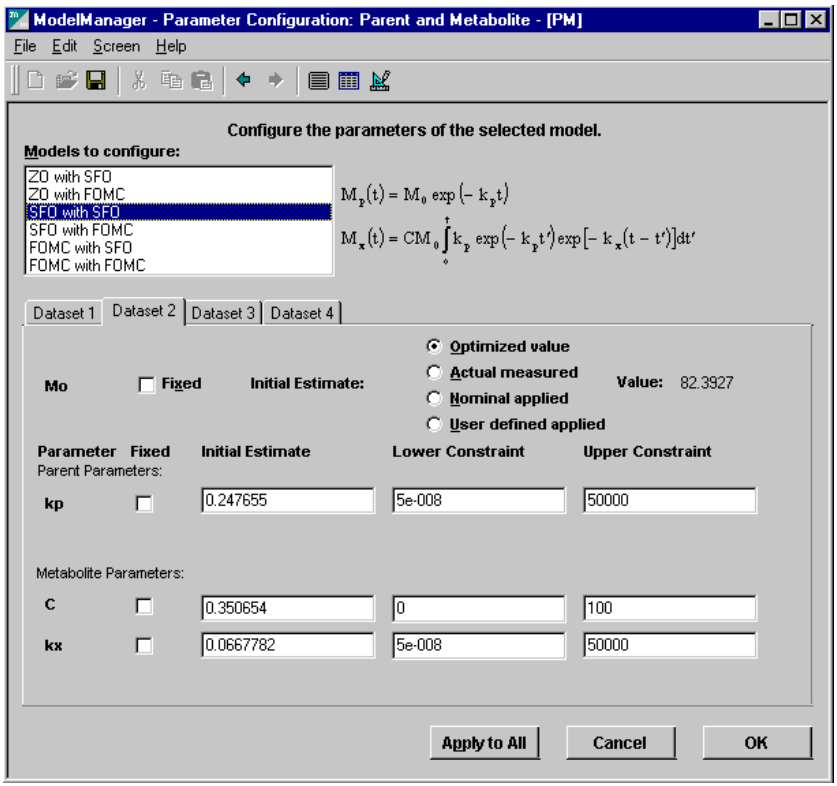
If a report is not desired during the current analysis, the check box under the heading "Generate Report during Analysis" should be de-selected. The user may create a report at any time (while an analysis is not currently running) by clicking the "Create" button. This button is not available until an analysis has already been run.

If a report has been requested to be generated automatically at the end of the analysis, and any of the model optimizations have not been successful, then the user is prompted, at the end of the analysis, as to whether they wish to proceed with report generation.

Configure Parameters Screen

This screen allows the user to alter the default values of a particular model. The parameters for a particular model may be set to different values for different datasets if required.

The Configure Parameters screen



Models to Configure

This list displays all the models associated with the study type, whether they were selected in the analysis configuration screen or not. The user should select the model whose parameter should be edited from this list. Only one model may be selected from this list at any time.

When a model is chosen from this list box, this causes the appropriate parameters to be displayed on the screen. In addition, the appropriate bitmap is selected to display the form of the chosen model.

Model bitmap

The model bitmap displays the equation associated with the chosen model.

Dataset Tab Control

The tab control may be used to switch between datasets, so that the model parameters that are used are dependent on the dataset. For example, the diagram above shows the parameter values to use when running the “First Order Multi-Compartment” model on dataset 1. If a Model / dataset combination is chosen, which was not selected from

the lists on the Analysis Configuration screen, then the screen entries are dimmed and may not be edited.:

M0 Properties

The M0 parameter is common to all models, except for those models included in the Parent with Multiple Applications study type. The value of M0 is specified from the options described in the list below. The user may specify whether the value is "Fixed" in the optimization - i.e. it is not included as an adjustable parameter in the optimization. The value of M0 which is to be used is selected from the four options (three options for Parent with Multiple Applications studies):

- **Optimized value**

  This takes the value which was calculated during the previous analysis. The actual value used is shown to the left of the radio buttons. This option is not available in Parent with Multiple Applications studies.

- **Actual measured**

  This takes the value which was entered on the experimental data entry screen, which corresponds to the entered value at time t = 0.

- **Nominal applied**

  This tells ModelManager to use the value entered on the experimental data entry screen under the heading "nominal

- **User defined applied**

  This tells ModelManager to use the value entered on the Experimental Data Entry screen (see Section 0) under the heading "User defined applied".

If the user selects a value for M0 that is not available, for instance if they select the Nominal Applied value, and have not specified this value on the Experimental Data Entry screen, then they will receive the message "Your initial estimate for M0 does not exist. Please select another.".

For the Parent with Multiple Applications study type, the application data entered on the Experimental Data Entry screen for each dataset are used as a series of fixed M0 parameters.

Parameter Properties

The parameters are configured by entering an initial value, a lower constraint and an upper constraint to be used in the optimization. The parameter may be taken out of the optimization process by selecting the check box associated with that parameter labeled “Fixe

“Apply to all” button

When this button is clicked, the parameter estimates and the “Fixed” status for the model currently on the screen are applied across all datasets. This includes the options for the M0 parameter and the constraint ranges of the parameters.

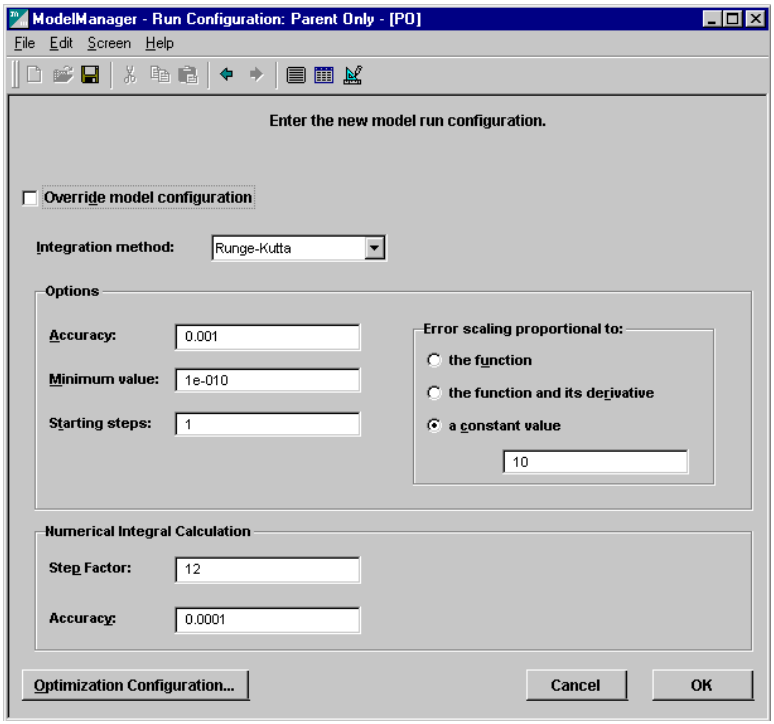
Run Configuration Screen

How to get to this screen

The Run Configuration screen may be opened from the Screen menu. This screen is password protected with “Super-user” privilege. Once a correct password has been entered, the screen shown below will be displayed.

The screen shows the run configuration that may be adjusted for the analysis. The integration schemes used by ModelManager uses a variable (not fixed) step length. A full description of the methods is beyond the scope of this document. Please refer to the ModelMaker user manual for details.

The Run Configuration screen



Override Model Configuration The entries on this screen will not take effect unless this check box is set.

Integration method This specifies which of the four integration methods to use. The available methods are Euler, Mid-point, Runge-Kutta and Bulirsch-Stoer.

Accuracy This specifies the accuracy of the integration method. Small values provide greater accuracy but cause longer run times.

Minimum value This is the minimum value allowed when calculating the model parameters. This is used to prevent numerical overflows during calculation.

Starting steps This is the initial number of divisions that the initial output step should be split into.

Error scaling This allows the user to specify what method is used for the error scaling. The choices are error scaling proportional to:

- the function
- the function and its derivative
- a constant value

See the ModelMaker online documentation for more details on the mathematics behind these options.

Step Factor This allows the user to specify the step factor to use for numerical integration during model calculation.

Integration accuracy This allows the user to specify the numerical accuracy for numerical integration during model calculation.

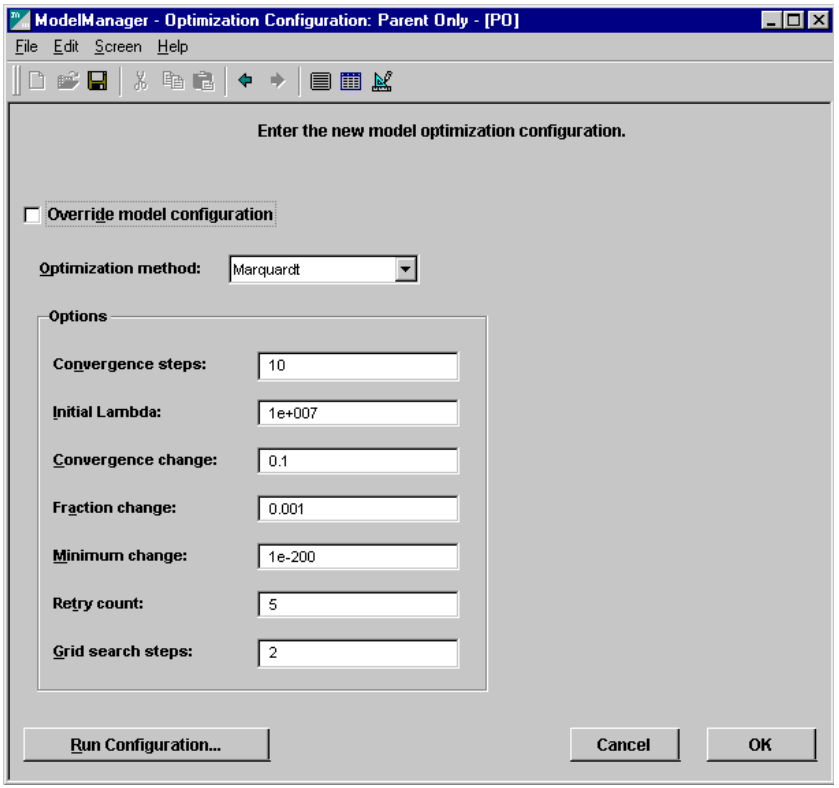
Optimization Configuration button The user may enter the Optimization Configuration screen by clicking this button. As the user has already logged in as Superuser, no further password is required. The Optimization Configuration screen is described in Section 0.

Optimization Configuration

This screen, also opened from the Screen menu, is also password protected, except when entered from the Run Configuration screen.

As for the Run Configuration, a full description of the fields in this screen is beyond the scope of this document. Please refer to the ModelMaker user manual for details. The elements of the screen are described below:

The Optimization configuration screen



Override Model Configuration The entries on this page will not take effect unless this check box is set.

Optimization Method The user may select from Simplex or Marquardt optimization.

Convergence steps This specifies the number of successive convergent steps that must occur before the model is considered to be optimized.

Initial Lambda This value is applicable only for Marquardt optimization, and is the parameter by which the Marquardt method moves between the methods of curvature and steepest descent.

Convergence change This is the fractional change in χ^2 which is considered to be a convergent step.

Fraction change This is applicable to the Marquardt method and controls the values calculated for the curvature matrix.

Minimum change This is also applicable only to Marquardt, and specifies the lowest tolerated value of the fractional change. See the ModelMaker manual for details.

Retry count This is the number of times a calculation is retried if either a parameter violates its constraint range, or if an optimization step is carried out without the parameter value changing.

Grid search steps This specifies the number of steps to use in the grid search. The larger this number the smaller the "mesh" used in the grid search.

Run Configuration button The user may enter the Run Configuration screen (described in Section 0) by clicking this button. As the user has already logged in as Superuser, no further password is required.

ModelXchange

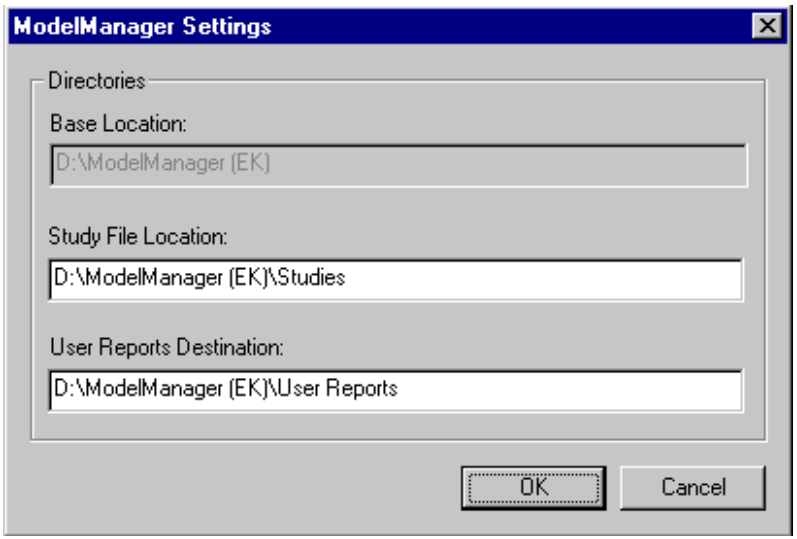
ModelXchange is an administrative function that allows a model set (i.e. the standard, weighted and log models) to be exported and imported through a single file. This functionality will be described in more detail later in this document.

Please note that Cherwell Scientific provides a web site www.ModelXchange.com for exchanging these files with other users of ModelManager.

Settings Dialog

The Settings dialog may be found under the Edit menu of the ModelManager toolbar.

The Settings dialog



- Base Location - this field is not editable by the user, and displays the installation directory of ModelManager.

- Study File Location - this field shows the default directory that will be displayed when the user opens an existing study in the opening screen of ModelManager. The user may alter this field if loading studies from a different folder.

- User Reports Destination - when ModelManager generates reports, these are saved to the directory named in this field. The user may alter this field to have the reports put into a different folder.

6. Constructing Reports

Introduction

This section contains information on setting up Excel to run ModelManager reports, how to enhance current reports and to create new reports.

Pre-Requisites

There are no assumptions about which add-ins are installed or selected on the end user's machine.

The only pre-requirement is that the DAO 3.0 Object Library is installed. To ensure that this library is available, the administrator should check the available *references* in the default system Excel installation. This is done as follows (this requires an open workbook):

- Open a new blank document
- select **Macro | Module** from the **Insert** menu
- select the new worksheet, named Module 1
- select **References** from the **Tools** menu
- scroll down the list and check that Microsoft DAO 3.0 Object Library is *included* in the list. It is not required that this is *selected*
- click OK or cancel to leave the dialog

Report Worksheets

The report workbook contains many different sheets to which data is written during report construction. All of these sheets are hidden during this process. In this section the main features of the worksheets will be introduced.

Open a template file using Excel. To view all the sheets in the workbook, first ensure the workbook is unprotected by selecting **Protection | Unprotect Workbook**... from the **Tools** menu. If this option is not available, then the workbook is not protected. Next, select **Sheet | Unhide**... from the **Format** menu, which brings up a list of hidden worksheets. Select the worksheet to reveal and click OK. Repeat this method to reveal all the hidden worksheets in the workbook.

The "General" Sheet

When the reports are generated, the calling application ModelManager has access to this page via DDE connection[2]. The important cells on this page as far as filenames are concerned, are those to which ModelManager must write information. The following information is written onto the "General" sheet in columns 1 and 2, starting at row 7.

Table 1:The relevant cells on the "General" sheet to which ModelManager writes information as part of report initialization

Version Details:	No Name
Bitmap Directory	<base directory>\Bitmaps\
Save directory	<base directory>\User Reports\
Save filename	<base directory>\User Reports\No Name.xls
Database Filename	<base directory>\Mman

ModelManager writes the name of the current study into the cell C2R7 (read Column 2 Row 7 - can also be referred to as cell B7), and the destination directory for the report into cell C2R10. The entry in C2R11 concatenates these two entries and appends the extension ".XLS". It is this cell that is used as the filename to which to save the

[2] Dynamic Data Exchange (DDE) is one of the interprocess communication mechanisms supported by Windows 95.

worksheet before constructing the report. The directory containing the bitmaps used in ModelManager is specified in cell C2R9.

ModelManager writes the database filename to the cell C2R14 of this sheet. This filename is used by the source code in the report to specify the source of data.

Data Retrieval Tables

The method of specifying which data fields should be fetched by the source code of the report is done via the data retrieval tables on the "General" sheet. The user will notice there are three such tables on the "General" sheet, the first starting in cell C2R22, the second starting in cell C2R29 and the third starting in cell C2R36.

It is important that the tables are not moved from the cells stated above, and that the cells **between** these tables (in rows 27 and 34) are left blank, or errors may occur when generating the reports. It is this blank row which is used by Excel to delimit the end of the table above it.

Each row of each table contains a list of indices that are used by the Excel macros to refer to entries on the "Filenames" sheet of the workbook. On the "Filenames" sheet, the user specifies lists of database tables and fields that should be retrieved from the database when constructing the report.

Each of the three tables on the "General" sheet specifies data to be fetched, with this data falling into one of three categories:

- Data relating to the entire study

  This is data that is relevant to the analysis as a whole, and is not specific to any particular model or dataset.

  Data falling under this category includes Study Details (study name, study user name etc), the analysis configuration (value at which to report DTX etc.) or the study type. Data requested in this table is retrieved only once during report generation.

- Data relating to an individual dataset

  This is data that is specific to a particular dataset, but is not specific to any particular model included in the analysis.

Information such as dataset name, nominal applied amount, time course data etc. differ between datasets. This information is retrieved for each dataset included in the model run, so these fields will be fetched the same number of times as there are datasets in the study.

- Data relating to a particular model fitted to a particular dataset.

  This is data that is specific to a particular dataset and a particular model.

  For example, if a model is fitted to several datasets, its parameters are likely to have different values for each of the two datasets.

If, during the model run, two models were selected for two datasets, then there will be two model fits stored for each dataset. Each dataset / model combination is referred to as a **Case**. The entries in this table are therefore run as many times as there are cases in the analysis. For example, there are four cases for two models fitted to two datasets. Example entries on this table are fit points, model parameters etc.

The columns of these three tables are described below. Note that of these values, it is likely that the administrator will only ever have to change the columns named “src record col”, “dest row”, “dest col”

- Label

  This field is a label to help the user remember which field on the “Filenames” sheet that particular row of the table refers to. This field is purely informative - it does not refer to any cell in the “Filenames” sheet and is not used in any way when importing data.

The table on the "General" sheet which specifies which information, that relates to the study as a whole, should be imported during report generation.

Label	source table row	source record row	sort by row	src table index row	src record col	dest row	dest col	dest page
Study Details	10	17	N / A	13	2	1	1	Details
Analy Config	10	17	N / A	13	42	23	2	Details
Study Type	10	17	N / A	13	57	1	8	Details

- Source table row

  This specifies the row number on the "Filenames" page in which the name of the database table is specified. This value is currently fixed at 10.

- Source record row

  This specifies the row number on the "Filenames" page in which the list of records to retrieve from the named table begins. This value is currently fixed at 17.

- Sort by row

This field is obsolete.

- Src table index row

  Many tables in the database contain indices which associate data values with a particular model or database, for example dataset ID or case ID. If the user wishes to match a particular index in the table with its equivalent in the case table (see section 0 for an example of the case table) then the user should enter the name of the index on this row of the "Filenames" sheet. For example, dataset names should be imported according to dataset ID.

- Src record col (source record column)

This is a column index, which identifies the column on the "Filenames" sheet in which the specification of a particular data fetch appears. For example, on the "Filenames" sheet, the specification for the fetch of "study details" appears in column 2.

- Dest row (destination row)

  This field specifies the destination row in which the fetched data is to be placed on the destination sheet.

- Dest col (destination column)

  This field specifies the destination column in which the fetched data is to be placed on the destination sheet.

- Dest page (destination page)

  This field specifies the destination sheet on which the fetched data is to be placed.

Note that if the format of the "Filenames" sheet is kept the same as it is at present, then the first three columns containing numbers (source table row, source record row, src table index row) will remain constant, and will never need to be changed.

Specifying the Offset between data imports

When data is fetched from the database, this data is pasted to a sheet within the report workbook. Because data may be retrieved for more than one dataset or model, then this data needs to be offset by a certain number of columns from the original data rather than overwritten on top of it.

Cell C9R19 of the "General" sheet is for the user to specify the column "offset" used in report generation. This parameter is necessary when data is being retrieved for several models or datasets. For example, if the study was done on more than one dataset, then this specifies the offset that Excel should leave between data associated with each dataset. It is not expected that the administrator would need to change this value.

The “Filenames” Sheet

This sheet contains the specification of all the data fields to be extracted from the database. It consists of a number of columns, specifying database table names, and the records to be selected from this table.

The fields that are to be collected from the database by the Excel macros are entered on this page under the headings “Database Table Records”. The Visual Basic macros are written in such a way that they take the field names from the list, which may contain any number (>0) of field names, and constructs a command which it uses to get the fields from the database. Note that there should be no blank rows within the main body of the list.

Other Sheets in the Workbook

The “Wait” sheet should be the only sheet not hidden in the final template report. All other sheets in the workbook are used as destinations for data imported from the database. The report will display the “Wait” screen while data is being fetched into the workbook, and then the newly constructed report will be presented to the user, and the template sheet closed down..

If a different opening sheet is required to be shown while report generation is done, the name of this sheet should be entered in cell C2R16 of the “General” sheet.

Important Note

The “General” sheet must be left unprotected in the template report, or the DDE connection from ModelManager (which writes to this sheet) will fail (with no error). It is recommended that this sheet is left unlocked but hidden in all templates to be opened by ModelManager.

Final Report Format

Presentation sheets

This is the name given to the sheets that are included in the template reports that are designed to display analysis results. Three “types” of report may be identified:

- Case by case sheets

These sheets may show the results for the fit of a single model to a dataset. The final report produces one of these sheets for each model selected, and names it accordingly.

- Case-by-case details sheets

  This sheet may contain a more detailed presentation of the results for the fit of a single model to a dataset. The final report produces one of these sheets for each model selected, when no comparison sheet is available.

- Comparison sheets

  These sheets may contain comparative information on the fit of several models to a dataset.

Any of the above sheets may be output to the final report. When more than one dataset is selected, the fit of the model to subsequent datasets is pasted horizontally adjacent to the previous ones on any particular type of presentation sheet. Therefore the system administrator need only design the above presentation sheets for one dataset.

For example if the user selects 2 models in their analysis, it is possible to output case-by-case sheets for both of the individual models to each dataset included in the analysis, and also a comparison sheet showing all model fits for each individual dataset superimposed on the same sheet. Or, if the user specifies only one model in their analysis, it is possible to replace the comparison sheet with a details sheet.

Delimiting Presentation Sheets

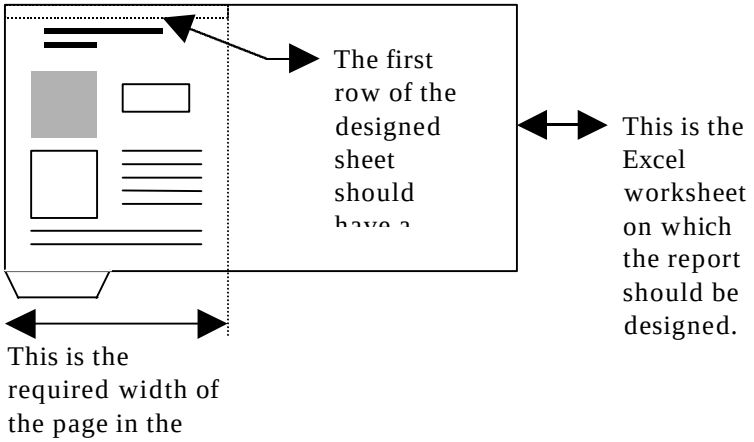
In order for correct behavior of Excel when generating the reports, it is necessary for the administrator to format the background of at least the first row of these presentation report pages. The width of the region to be copied and pasted, for separate datasets, into the output report should be delimited by the background of this area being of solid color. In order to do this, select the required width of the area to copy in row 1 of the sheet, starting from cell C1R1. Now click the right mouse button and select **Format Cells…** from the list. On the tab labeled **Patterns**, select any color from the grid labeled **Cell Shading**. White is recommended as it will not show up on the final report when printed out.

Formatting text on Presentation Sheets

When designing the presentation sheets, all text formatting will be preserved when the text is pasted across to the new report. For example, Font style, Bold, Italic and underline formats, column and row widths etc.

Note however, that superscript and subscript effects are **not** preserved, when these have been specified using **Format|Cells...** from the Excel toolbar. In order to incorporate superscripts and subscripts, it is advised that the administrator use the "Character Map" of Windows. See the Windows documentation on the method of opening and using the character map. These character formats **are** preserved in the output report.

Schematic diagram of the region of the presentation sheet that should be delimited with white space as described in the text



Other rows in the presentation sheet region may also have solid color backgrounds, but only for the first row is this mandatory.

Configuration of the Output Report

The format of the final report that is displayed to the end user may be configured by the administrator via fields on the "General" sheet. The number and format of the report sheets in the final report may depend on the number of datasets and models selected for analysis by the user. The following cells control this configuration.

Table 3 Area of the General sheet in which the administrator should specify which sheets should be exported to the final user spreadsheet, according to the number of models that were selected in the analysis

<cell C5R6>	1 Model	2 Models	3 Models	4 Models	
Comparison sheet name:	1 Model Detail	2 Model Comparison	3 Model Comparison	4 Model Comparison	
Case-by-case sheet?	TRUE	TRUE	TRUE	TRUE	cell:
Use Bitmaps?	TRUE	TRUE	FALSE	FALSE	B2
			Case-by-case sheet name:	1 Model	
					cell:
			Use bitmaps?	TRUE	J11

This table starts in cell C5R6. The cells C6R7 to C9R7 specify the Excel comparison sheets to use, according to whether the user included 1, 2, 3 or 4 models in the analysis.

Important: The administrator should ensure that all worksheets referred to on the "General" worksheet actually exist, e.g. if you specify the sheet to be displayed for a 2 model comparison as "2 ure that the template workbook contains a sheet with this name. Any errors made in this respect will be obvious when the administrator tests the new template report.

If the administrator does not wish to allow e.g. a comparison of 4 models on a single sheet, then leaving the appropriate cell blank (cell C9R7) will tell the report macro that comparison reports of this type are not allowed. The macro will then generate reports containing only presentation sheets on an individual model basis, if 4 models are run.

The "General" sheet allows the administrator to configure the comparison sheets (of up to 4 models). If the user runs analysis of > 4 models, then by default only individual model sheets will be produced.

In addition to comparison sheets, showing comparative model fits to the same dataset, the user may also wish to see the individual fit of

each model to each dataset. For this to occur, the administrator must have set cells C6R8 to C9R8 appropriately. If these cells are set to TRUE then these individual model sheets will be generated. These cells should be set to FALSE if these sheets are not to be enabled.

Parent and Metabolite Model Pairing

As a special case, some extra handling has been included in template reports for Parent and Metabolite studies. It was required that these study types only show 2-model comparison sheets if the user requested a particular pair of models. If the models selected by the user did not constitute a pair, then individual model case-by-case sheets are generated for each of the models.

In order to accommodate this, an extra table has been included on the "General" sheet for the administrator to specify how the models should be "paired". Note that it is important that associated models appear next to each other in the model list in ModelManager, i.e. models 1 and 2 may be associated, but models 2 and 5 may not.

This table begins in cell C18R9 of the "General" sheet, and appears as shown below:

Table 4 Table specifying model pairing for the Parent and Metaoblite study type

1st Model	2nd Model	Comparison name
1	2	ZO Parent Comparison
3	4	SFO Parent Comparison
5	6	FOMC Parent Comparison

Note that it is important that sheet names specified are below 31 characters in length.

The two models which should be paired with one another appear in the same row. The example above pairs models 1 with 2, 3 with 4, and 5 with 6. The sheet name that should be used for the comparison sheet of these two models is specified in the 3rd column.

If the two models do not form a pair, then sheets showing detailed information on each of the models may be included instead. Cell C9R11 allows the administrator to specify which sheet should be included instead of the 2-model comparison sheet.

The name of this details sheet in the final report will be of the form "<Model Name> <Detail Sheet Name>", for example if the detail sheet is named "Details" then in the final report, the sheet showing details for the ZO-FOMC model will be called "ZO-FOMC Details".

Note that if the user selects e.g. all 6 models, then these will be grouped into 3 groups of 2 models, displayed on three 2-model comparison sheets.

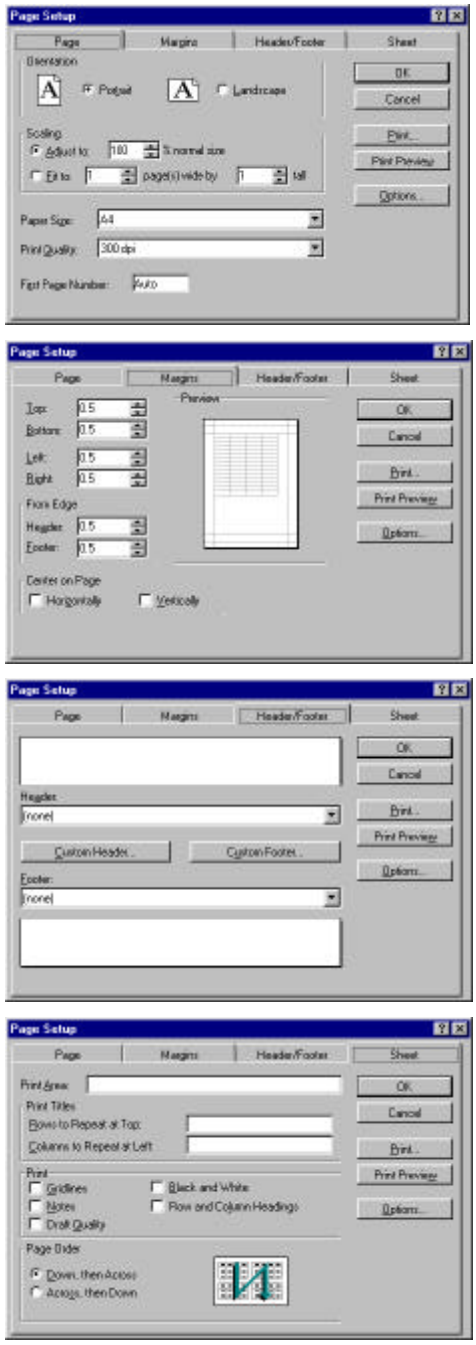
Format of Presentation Sheets

By presentation sheets, we refer to pages in the template book which have been designed to show the analysis data and results, and that are to be pasted into the new book. These sheets may contain any number of chart objects.

It is recommended that the background of the active sheet area be made a solid color, e.g. white. If this is not desired, then it is **very important** that at least the **first row** of the sheet is of solid color, across the desired width of the sheet. The Excel macro which copies and pastes the sheet from the original template book to the end book determines the width of the sheet to be pasted by the number of adjacent solid color cells in the top line of the sheet to be copied, starting at cell C1R1. This is important for when there are several datasets included in the analysis. In order to show the model fits for several datasets aligned side by side in the output report, the macro needs to be able to determine the width of the sheets it is aligning.

The size of the sheet in the template workbook should be of the same size as that required in the final report. Headers and Footers are not included in the final sheet, and so the sheet should be of suitable size to fit on A4 page size you wish using File | Print Preview. Text, column and row formatting are preserved in the export over to the presentation report.

Diagrams showing the print settings that will be used as default in the final sheets created by ModelManager template reports.



The user may change these settings once the report has been generated. These settings are not protected in the final book.

Bitmaps

ModelManager reports offers the means to import model bitmaps into sheets. The bitmaps that are imported are those bitmaps currently associated with the model (i.e. those bitmaps that appear on the parameter configuration screen of the ModelManager user interface).

The method of specifying whether bitmaps should be included on comparison or individual model sheets is as follows:

- Comparison sheets

  On the “General” sheet, the administrator should set the cells C6R9 to C9R9 appropriately, to specify whether bitmaps should be included on comparison sheets of 1, 2, 3 and 4 models. If bitmaps are to be included on multiple model comparison sheets, then the bitmaps are arranged immediately beneath each other, for example, on the 2-model comparison sheet for the Parent and Metabolite models ZO with SFO and ZO with FOMC the arrangement of both bitmaps will be:

$$M_p(t) = M_0 + m_p t$$
$$M_x(t) = -\frac{Cm_p}{k_x}\left[1 - \exp(-k_x t)\right]$$

$$M_p(t) = M_0 + m_p t$$
$$M_x(t) = -\frac{Cm_p b_x}{1 - a_x}\left[\left(\frac{t}{b_x} + 1\right)^{1-a_x} - 1\right]$$

Note that borders are not used in these bitmaps. The end user may choose to add borders to these bitmaps once their final report has been generated[3].

- Individual model sheets

  If the administrator wishes bitmaps to be placed on individual model sheets, then they should enter TRUE in cell C9R13 of the "General" sheet.

- Placement of Bitmaps

  The administrator should specify the cell where the bitmaps are to appear. To specify the position of the bitmaps on case-by-case and comparison sheets, the cell name should be entered in cell C10R13 of the "General" sheet. For the comparison sheet bitmaps, the cell name should be entered in cell C10R9. In both cases, the cell name entered should be of the form e.g. "A1" and not "C1R1".

Visual Basic Macros

The Visual Basic source code of the macros which construct the report is included in the Module sheet Module1. The code has been written to make it as simple as possible to modify the behavior of the macros without having to alter this source code. In order to do this, many of the adjustable variables that are used in the code have been "exposed" by including them on the "General" and "Filenames" sheets. This means that to modify the behavior of the reports, it should not be necessary for the administrator to modify the source code at all.

One of the main functions of the macros is to create an SQL[4] statement from the fields specified on the "Filenames" worksheet. This SQL

[3] As graphical objects such as charts and bitmaps are not password protected in the final user-generated reports.

[4] Structured Query Language (SQL) - used for database access.

statement is passed to Microsoft DAO functions to import the data. The full SQL statement may be rather long.

There are many macros in the report spreadsheets supplied with ModelManager. Some of the most important macros, which should always be made available in future reports are described below. The macros indicated by an asterisk are vital as they are executed externally by ModelManager and are therefore essential for the correct execution of the report.

- *GenerateReport()

  This is the main macro of the workbook, and calls all other macros in the workbook which import data. In addition it performs other necessary functions, such as disabling user termination of the running macro, exporting presentation sheets out into the new workbook etc.

In addition to the above named macro, there are a number of *functions* on the Module sheets[5]. These functions are not available to be run by the end user individually. Three important functions (those that import data to the template spreadsheet) are listed below:

- ImportGeneral(....)

  This function is called by the `GenerateReport` macro, and loads data relating to the study as a whole, such as study details, Analysis configuration etc.

- ImportDataSet(....)

  This function is called by the `GenerateReport` macro, and is used to import data that is indexed by the dataset ID, such as the time course data, dataset details etc.

- ImportData(....)

[5] Functions differ from macros (subroutines) in that they are able to receive values as arguments, and return values.

This function is called by the `GenerateReport` macro, and is used to import data which is indexed by case ID, such as model fits, model parameters, model fit points etc.

Pseudocode of Report Generation Macros

This section will outline the structure of the visual basic code that is behind the generation of reports in ModelManager. It should not be necessary for the administrator to make alterations to this code.

Case ID

The concept of case ID is as follows: If the user selects 2 models and 2 datasets for analysis, then both of the models will be applied to both datasets. We therefore have 4 "cases" in our analysis corresponding to cases:

1. Model number 1 applied to Dataset number 1
2. Model number 2 applied to Dataset number 1
3. Model number 1 applied to Dataset number 2
4. Model number 2 applied to Dataset number 2

On initialization of the report, the ModelManager "case table" is imported into the report, and is used by the report as a basis for report construction. An example of such a case table is illustrated below, for the analysis of two models (Models 1 and 2) to four datasets (datasets 1, 2, 3 and 4) in study type 2 (Parent and Metabolite):

CaseID	modelID	datasetID	StudyTypeID
1	1	1	2
2	2	1	2
3	1	2	2
4	2	2	2
5	1	3	2
6	2	3	2

7	1	4	2
8	2	4	2

Table 1 Example of the case table which is used by the template report to generate the output report.

The table above illustrates the case table for two models fitted to four datasets. The models each have an ID number, modelID. In the case above the models have ID 1 and 2 respectively. These are both applied in turn to each dataset. The datasets themselves have an ID number, datasetID.

Also imported separately are the numbers of datasets and models, and the list of model names and their bitmap names.

It can be seen that the running index caseID corresponds to a particular modelID / datasetID pair. For example, caseID = 4 represents model with ID = 2 applied to dataset with ID = 2.

Clearly we wish to import four datasets, and eight model fits. The Visual basic code which constructs the report initially imports values for the total number of models and the number of datasets selected in the analysis. These totals are 2 and 4 respectively for this example.

The code then progresses through the case table and calls the functions to import data according to the following pseudocode:

```
<go to first row of case table>

For I = 1 to number_of_datasets

ImportDataset(....)     ‘ (retrieve dataset data
etc.)

      For J = 1 to number_of_models

            ImportData(....)  ‘ (retrieve model fit
data etc.)

            <increment case table>

      Next J

<Export presentation sheets to new workbook
```

```
                                        for this
                                    dataset>

Next I
```

Note that the appropriate offset, described in section 0, is made each time to the destinations for the imported data.

Before this loop is started, the function ImportGeneral is executed once, which imports all the study details etc. which are associated with the study as a whole. An example of importing data is given in section 0.

Example: ImportGeneral(…)

We will illustrate the way database record names are specified and used for a data query. On the "Filenames" sheet, the following cells appear in column 2:

The specification in the "Filenames" sheet of the study details data fetch.

STUDY DETAILS (General)
Identifying Features
Order by
Database Table Name
StudyDetails
Indices (max 2)
Database Table Records
Study DOA
StudyName
StudyNumber
StudyUserName
StudyDesc

The location of this specification is inserted into the table on the "General" sheet, which contains the following line:

The row of the table on the General sheet which specifies the location and destination of the data import of the Study Details of the analysis

Label	source table row	source record row	sort by row	src table index row	src record col	dest row	dest col	dest page	off set
Study Details	10	17	N/A	13	2	1	1	Data set	6

This specification is read by the `ImportGeneral()` function, which produces the following data table on the "Dataset" sheet, which contains all five fields specified on the "Filenames" sheet, arranged horizontally, with the corresponding information pasted beneath:

The data table produced by the ImportGeneral() function

StudyDOA	StudyName	StudyNumber	StudyUserName	StudyDesc
1998-02-23	Study1	001	Tim	Field study

The administrator may append to the specification list on the "Filenames" sheet, any new database records to be extracted from the specified table. The macros which reads this list will then pick up this additional name and fetch the additional record(s).

Note that the list of records specified do not have to be in any particular order. However, note that the order in which they are specified is the order in which they will be pasted into the "Dataset" sheet.

The meaning of the field "Order by" is as follows: suppose we are retrieving data that should be in a certain order, for example time course data should be ordered by time when fetched from the database, before being pasted into the template workbook. To do this, the administrator should enter in row 5 the name of the record according to which the data should be ordered. Note that this record does **not** have to also appear in the list of table records that begins in row 17.

The database fields available in ModelManager are listed in a separate document. However, it may be informative to open up the database and examine the fields during this process. See the separate documentation on ModelManager databases for information on opening the password-protected databases using Microsoft Access.

Example: Adding Records to an Existing Macro

Suppose we wish to add information of the number of datasets in a particular study. This information exists in the ModelManager database in the table named StudyDetails, under the field name StudyNumDatasets. To have this data retrieved by the macro `ImportGeneral()`, we add the field to the list shown above, so we have the table below.

The amended specification in the "Filenames" sheet for ImportGeneral(). The appended field is at the bottom of the table

STUDY DETAILS (General)
Identifying Features
Order by
Database Table Name
StudyDetails
Indices (max 2)
Database Table Records
Study DOA
StudyName
StudyNumber
StudyUserName
StudyDesc
StudyNumDatasets

After the `GenerateReport()` macro has been executed with this change, the following information will be written to the "Dataset" worksheet.

Data table produced ImportStudyDetails( ) with the amended specification.

StudyDOA	StudyName	StudyNumber	StudyUserName	StudyDesc	StudyNumDatasets
1998-02-23	Study1	FRED001	Tim	Field study	7

Note that now the newly added record is appended to the right of the other records.

Displaying the New Information

This value can then be easily added to the presentation sheets of the report. This is done by selecting a cell on the presentation sheet on which the information should be included, and entering the cell number that contains the new information. For example, if this newly added cell containing the StudyNumDatasets value ("7") is cell C6R2 on the "Dataset" sheet, then the following should be entered into the desired cell on the presentation sheet:

```
=Dataset!F2
```

This will copy the new information to the cell on the presentation sheet. A number of functions are available to the administrator in making this link. Examples include the `IF`, `ISBLANK` and `SUM` functions.

The form of the `IF` function is a follows:

```
IF(<tested condition>, <action if true>, <action if false>)
```

The `ISBLANK` function takes a cell reference as an argument, and returns true or false according to whether the cell is blank or not.

For example, if we wanted to display the property StudyUserName only if it is not blank, then we might enter the following line into the presentation sheet:

```
=IF(ISBLANK(Dataset!F2),"",Dataset!F2)
```

The line above will set the cell of the presentation sheet to "" if the cell `Dataset!F2` is blank. The default behavior of Excel is to set this destination cell to "0" so this method is often necessary.

Note that when constructing graphs on the presentation sheets, these graphs should not refer to data contained in cells on the presentation sheets that have been copied across in the method described above. The graphs should refer to the original cells. So in our example above, if we wished to plot a graph of time course data that has been imported to the "Dataset" sheet, we should refer the graph to the cells on the "Dataset" sheet, and not those values that have been copied across to the presentation sheet. When done in this way, blank cells will be ignored by the graph.

An exhaustive list of available formatting options and functions is beyond the scope of this document. See the Excel online help facility for further information.

It may be useful for the administrator to enter "dummy" data at the position at which they expect the imported data to be placed, which might assist in constructing the presentation sheets of the template report. This dummy information will not appear in the final output reports from the template.

Example: Accessing New Tables

The ModelManager reports have been designed in such a way that the addition of completely new queries to new tables, rather than modify those already there, is as straightforward as possible. *No changes to the Visual Basic code are required.*

More information on the syntax of SQL statements is given later in this section, but here we give an example of what may be achieved simply by copying and altering an existing macro.

We already fetch the model name and the model parameter values in the template reports. Suppose we now want to add information on the integration and optimization techniques used in the analysis. These values are entered on the database table named ModelOptions. In the ModelManager working database mman.mdb, this table appears as shown below

The database table ModelOptions as it appears in the database

ModelOptions : Table

	RunIntegMethod	OptimMethod
	Runge-Kutta	Marquardt

This data relates to the study as a whole, and is not particular to any dataset or model. We therefore may retrieve this data in the same way as we would study details. To fetch this data to the current list:

- Go to the sheet labeled "General".
- In the upper of the three tables on that sheet (starting in cell C2R22) add a line to the bottom of the table, so the table is as below:

Amended table on the General sheet which includes the new data query to the ModelOptions table of the database.

Label	source table row	source record row		src table index row	src record col	dest row	dest col	dest page
Study Details	10	17		13	2	1	1	Details
Analy Config	10	17		13	42	23	2	Details
Study Type	10	17		13	57	1	8	Details
Model Options	10	17		13	62	1	10	Details

The “Model Options” specification on the final row of the table above means that we have chosen to put the retrieved data into row 1, column 10 of the “Details” sheet, as we have set “dest row” to 1, “dest col” to 10 and “dest page” to “Details”. The specification of the fields to import is defined as being written in column 62 of the “Filenames” tab, as we have set the value of “src record col” in the above table to be 62.

Next, in column 62 of the “Filenames” sheet we must specify what we want to be fetched by our new macro. That is, we specify the database table name from which the data is to be read, and the list of required records by adding the following to column 62 on the “Filenames” sheet (once this sheet has been unprotected):

The data specification for our data query of Model Options.

STUDY DETAILS (General)
Identifying Features
Database Table Name
ModelOptions
Indices (max 2)
Database Table Records

RunIntegMethod
OptimMethod

Ensure that the table name “ModelOptions” appears on row 10, and the first of the table records “RunIntegMethod” appears on row 17.

Once the report has finished fetching data, the following data will be found in cell C10R1 (row 1 column 10 - cell J1) of the “Details” sheet:

The data table fetched from the ModelOptions table by our new data fetch

RunIntegmethod	OptimMethod
Runge Kutta	Marquardt

Note that the actual *values* that you retrieve (i.e. “Runge Kutta” and “Marquardt”) may differ from those shown, if you ran your last analysis with other options than these. If the table was not fetched then check that the “Details” worksheet is not protected, and that the specification on the “Filenames” page is in the correct column and is correct, and that the table on the “General” sheet is correctly configured.

The administrator should set up the new table in the manner shown above. When they then create a link from their presentation sheets to this cell, this new information will appear on presentation sheets created by the template report.

Note on Debugging

If the administrator is familiar with the Excel programming environment, they may wish to make changes to the source code, or to step through this code to gain familiarity with it. If they do so, it is important that before the debugger is started, the administrator should comment out the line which disables user cancellation of the code.

This line appears in the source code as:

```
Application.Interactive = False
```

This line may be easily found in the code by selecting Find… from the Edit menu of the toolbar in Excel, and searching for this string.

If the administrator does not comment out or remove this line, the debugger controls such as step over, step into, stop debugging etc. will not be available, as the application is no longer interactive when the debugger reaches the breakpoint. In this situation Excel should be closed down.

Once the administrator has finished debugging the application, the command should be re-enabled so that methods of cancellation are disabled once more.

Worksheets

On the "General" sheet, there is a table starting in cell C21R22 (U22). This table specifies a list of worksheets that are to be *cleared* before data is imported into the report template. All sheets to which data is to be written should appear in this list.

In cell C2R16 (B16) the user should specify the name of the sheet which should appear while the report is being created. At present this sheet is the "Wait" sheet. If the administrator wishes to change this sheet, then the name of the new sheet should be entered in this cell.

Passwords

Please note the following points about passwords and Excel:

- The sheets in the final report generated by the template report are password protected. The password that should be used to protect and unprotect these sheets should be specified in the cell C2R20 (B20) on the "General" worksheet.

- When work*sheets* are protected, this prevents both the user *and Excel macros* to write to these sheets. Hence all sheets in the template reports should be left *unprotected* once the administrator has finished their design.

- When work*books* are protected, this preserves the structure of the book, so worksheets within that workbook may not be moved, deleted, hidden or un-hidden. It is recommended that the administrator protect the template work*book* against changes to its structure.

- It is highly recommended that passwords are stored in a safe place, as worksheets and workbooks cannot be unprotected in any way if their passwords are forgotten or lost. For the same reason, up-to-date backups of the template report workbooks should be maintained.

Checklist

The following points should be addressed before using a modified report:

- It is advisable that the template workbook structure is protected. This is done by selecting **Protection → Protect Workbook**... from the **Tools** menu when the template workbook is open. The administrator should enter a password to protect the workbook.

- Ensure ***all worksheets*** in the workbook are ***un***protected.

- It is important to store passwords, as protected workbooks may not be unlocked if a password is forgotten.

- Note that passwords are case sensitive.

- Make sure the cell C2R20 (B20) of the "General" sheet contains the password that you wish to protect the sheets in the workbook generated by ModelManager template reports.

- Ensure that all worksheets referred to on the "General" worksheet actually exist, e.g. if you specify the sheet to be displayed for a 2 model comparison as "2 Model Comparison Sheet", ensure that the template workbook contains a sheet with this name. The names should be identical in terms of spacing and upper/lower case.

- The administrator should keep backup copies of all template workbooks.

- The presentation sheets in the template workbook should contain at least a single row of solid color in the top row of the sheet.

- It is advised that all worksheets in the template workbook are hidden, with the exception of the "Wait" sheet (or, if the

administrator has specified that a different sheet is to be displayed during report generation, then this alternative sheet should be the only sheet not hidden).

7. ModelManager Wizard

Starting the Wizard

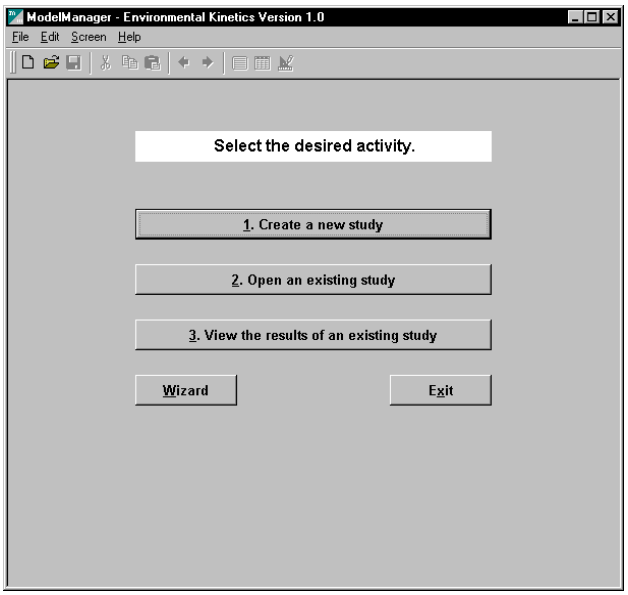
ModelManager has the facility for the user to add new reports and models to those already included with the application, or to delete existing reports and models.

The wizard does not provide an environment by which new models and reports may be *created.* New models should be created using ModelMaker, new reports using Excel.

In order to do this, the user must have security clearance to Supervisor level. Logging on as Supervisor is done by selecting "Supervisor mode…" from the "Edit" menu of the toolbar, and entering the correct password in the dialog box that is presented to them.

Access to the Wizard is via the opening screen of ModelManager, and when the user has entered the correct Supervisor password, an extra button appears on this screen as shown below:

The opening screen of ModelManager in Supervisor mode

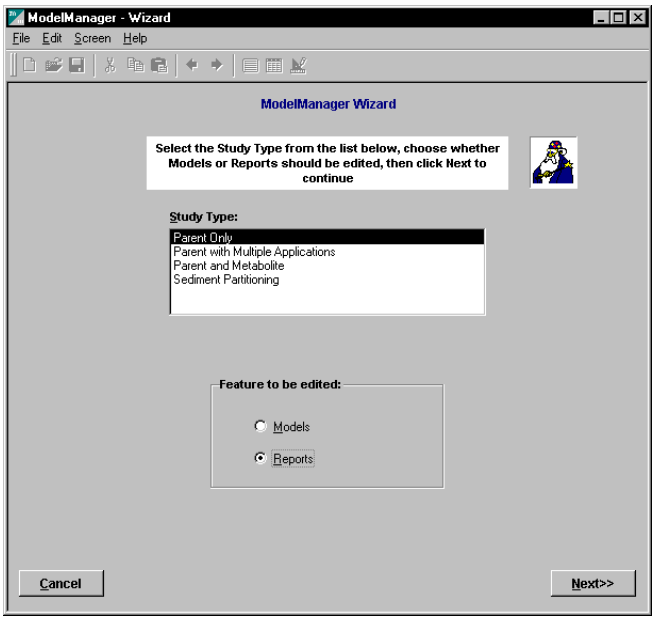


Clicking the button named “Wizard” will take the user into the Wizard facility. When the user returns to User mode, by selecting “User mode” from the “Edit” menu, the “Wizard” button will disappear, removing all access to the Wizard by the user.

Opening Screen

Clicking the Wizard button on the opening screen produces the ModelManager Wizard initial screen :

The opening screen of the ModelManager wizard



From this screen the administrator should select the study type to edit, and which of the two options should be taken:

- Model deletion or inclusion
- Report deletion or inclusion

In both cases, any changes made by the administrator will not take effect until they are confirmed on the final screen of the wizard. At any point in the report or model addition / deletion process, the administrator may quit the process without committing any changes. This is done via the “Cancel” button on each Wizard screen. The administrator will be asked to confirm their exit with the following dialog:

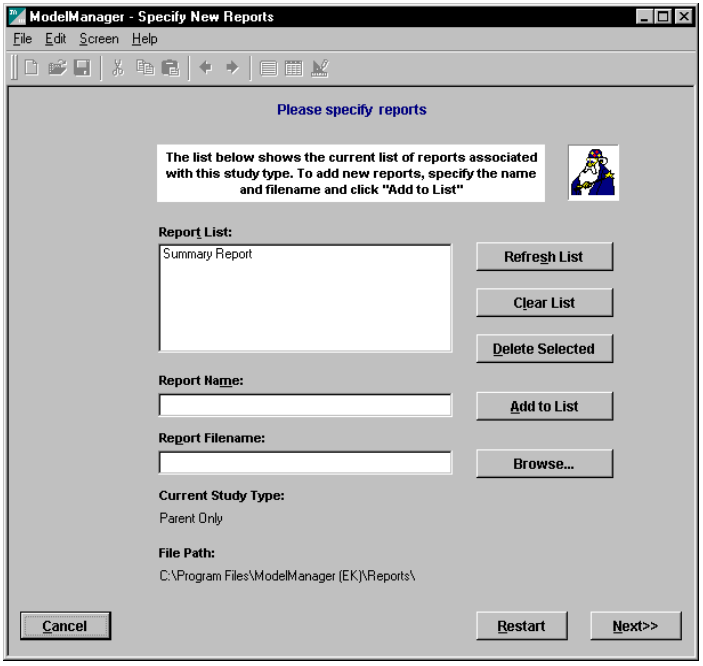
The confirmation dialog when the user cancels from the Wizard

Note that in many of the screens in the ModelManager Wizard, there are buttons labeled "<<Back" and "Next>>". These are to be used for basic navigation to the previous and next screens respectively in the user interface of the Wizard.

Adding Reports

When the administrator chooses to delete existing or include new reports, they are presented with the Specify New Reports screen:

Wizard screen to specify new reports



An explanation of the features on this screen follows:

- Report List - a list of currently available reports for this study type.
- Report Name - Edit box for the administrator to include the name of the new report that is to be added. This name is the name that will appear on the analysis configuration screen when the end user is performing their

analysis for this study type.

- Report Filename - This edit box is to specify the name of the template report file that is to be included. These report files should be in the directory shown under the heading “File Path” on this screen. The administrator may click the “Browse…” button to bring up a dialog showing all reports in this directory.

- “Add to List” button - This button will add the current report specified in the two edit boxes to the list of reports. This will only be done if the entries are valid and the filename exists.

- Refresh List - This button will restore the list of reports back to the list shown on first entry to the screen - i.e. to show the current list of reports included with ModelManager for that study type.

- Clear List - This button removes all reports from the list.

- Delete Selected - This button will delete all reports from the list currently selected by the administrator. The remaining reports will be re-ordered.

Once the list is configured as desired, the administrator should click Next>> to go to the final confirmation screen.

Adding Models

To follow the path through the wizard to add new models to a study type:

- Select “Models” from the ModelManager Wizard screen

- Select also the required study type from the list

- Click Next>>

This opens ModelFiles screen, described below.

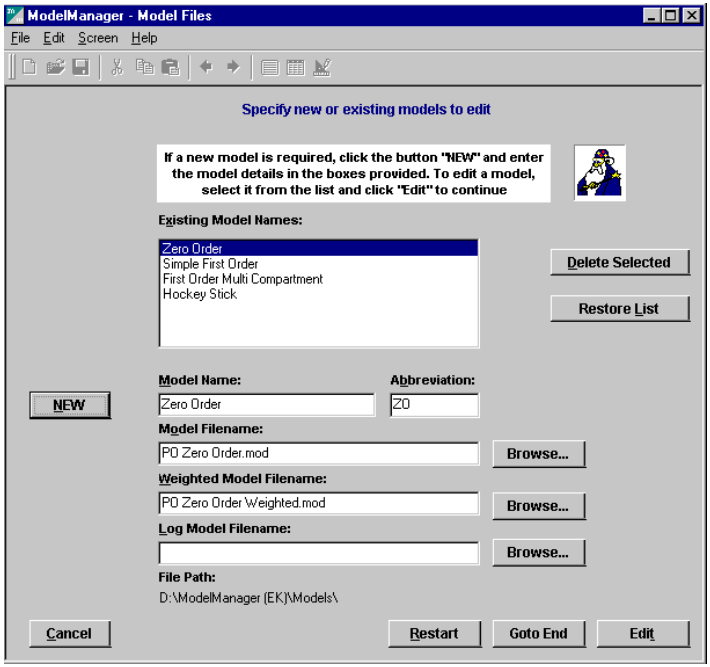
NB: When configuring a model using the wizard screens, the administrator will be presented with lists containing properties of the model, from which they may select their preferred choices. For example, the lists of events shown in the Parameter Search screen.

Lists such as these are obtained from the *non-weighted* model file. It is assumed that the weighted model will contain exactly the same model parameters and events, but with a different weighting configuration. If the two models do not contain identically named components and parameters, unpredictable effects may occur when running the model, or the model analysis may deliver incomplete results.

Model Names and Filenames

The Model Files screen is used by the administrator to specify the model details. Below we describe the functions of the individual buttons and sections of this screen.

The Model Files screen



This screen contains the following controls:

- Existing Model Names — This is a list of models currently contained within ModelManager. By clicking on a particular list item, the model details will appear on the screen, which may be edited.
- Model Name — This is the name of the model as it is to appear on the analysis configuration screen and reports.
- Model Abbreviation Name — In constructing the reports, it is useful to use an abbreviation to label the model, rather than its full name which might be rather long. This field is where the abbreviation should be specified.
- Model Filename — This is the ModelMaker model file that is to be used in the analysis. This model should be present in the directory shown on the screen under the heading “File Path”.
- Weighted Model Filename — This is the name of the ModelMaker file that is to be used when the user wishes a weighted model to be used in the analysis. This should be a ModelMaker model file in the directory shown on the screen under the heading “File Path”.
- Log Model Filename — This is the name of the ModelMaker file that is to be used when the user wishes an analysis to be performed on the log transform of the data. This should be a ModelMaker model file in the directory shown on the screen under the heading “File Path”. Note this file does not have to be added to the model set (e.g. it makes no sense to log transform zero order data)
- “NEW” Button — This button is used to start the process of adding a model to the list. When this button is clicked, a new model will be added to the list, and the Edit boxes will be filled with default settings. The user should complete these edit boxes (i.e. Model Name, Filename etc.) and then edit the new model by hitting the button “Edit”.
- Delete Selected — When this button is clicked, the model currently selected in the Model List will be deleted from the list.

- Restore List — This button will replace the contents of the Model list with the list of models currently available in ModelManager. Any models that have been added to the list will be deleted.
- Edit Button — Once the administrator has finished on this screen, clicking this button will take them to the next stage in the process of editing the selected model in the list. The model that is selected in the model list is that which will be edited.
- Restart Button — This will discard any changes to the model list and take the administrator back to the opening screen of the Wizard.
- Browse Buttons — These allow the user to bring up a dialog through which they may browse for the linear and logarithmic model files.
- End Button — If the user wishes to only delete models from the list, or merely to make changes to the model names and files etc. on this screen, they may click the “End” button to take them to the final confirmation screen once their changes have been made. This will bypass the remaining Wizard screens in the model configuration, which are not necessary for model deletion.

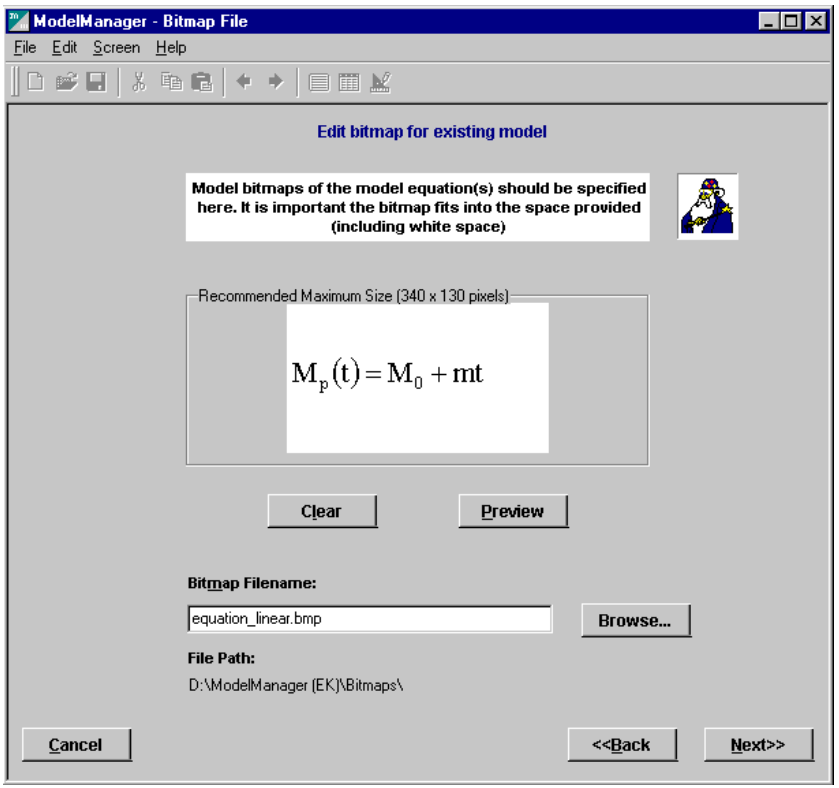
When the user exits the screen, the model that is highlighted in the list of models will be the model that is to be configured. If they have specified a new model on this screen, then they will not be allowed to configure any other model than the new one. If they have not specified a new model, then they may configure any of the models in the list.

If the user has edited any of the model details in the list on this screen, e.g. edited their names or filenames, all changes will be retained when the user commits the values on the final wizard screen.

Model Bitmap

The Bitmap File screen provides a means for the administrator to associate a bitmap with the model. This bitmap will then be available for display on the parameter configuration screen for that study type. It will also be available for display in the reports.

The Bitpmap File screen



The screen contains the following controls:

- Bitmap Filename

  This is where the administrator specifies the name of the bitmap file. The bitmap should be contained in the directory shown on the screen under the heading “File Path”. The button “Browse” enables the administrator to inspect the contents of this directory and select the file required.

- Preview Button

  This button will display the bitmap that is typed in the filename edit box. If the file has been selected using the “Browse” button, the bitmap will be automatically displayed.

- Clear Button — This will clear the preview area of the screen, and the filename edit box. The user may then specify another bitmap.

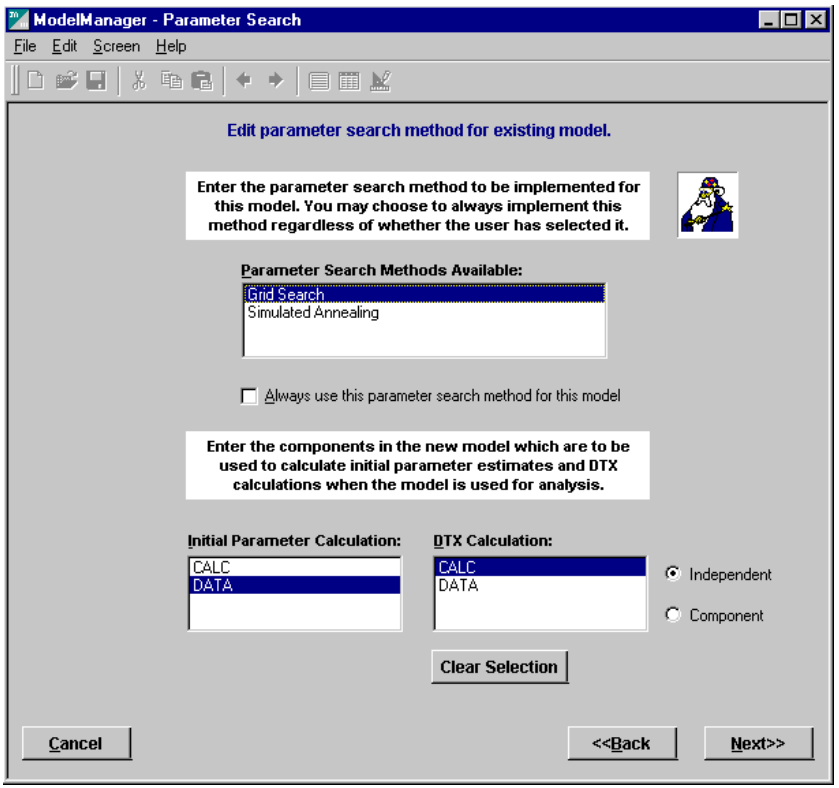
- Preview Area — This is the area in which the bitmap will be displayed so that the user can check it is the correct file, and that it fits into the recommended area (340 x 130 pixels). In the preview, the bitmap background appears white so that the true size of the bitmap is apparent. This white background will appear transparent when the bitmap is displayed on the parameter configuration screen.

Bitmaps should have a *white* background, for them to appear correctly on the parameter configuration screen. ModelManager renders all white pixels in the bitmap as transparent. The bitmap itself need not be monochromatic, and may contain more than one color.

Parameter Search Configuration

This screen enables the administrator to configure the method to be employed when the end user performs a parameter search to find the initial parameter estimates for this model. The parameter search configuration screen is shown below:

The Parameter Search screen - used to configure the initial parameter search and DT calculation



This screen contains the following controls:

- Parameter Search Methods List

  This list contains the current list of parameter searches available in ModelManager. The administrator should select one of these methods. There are two method currently available.

- Initial Parameter Calculation

  ModelManager contains a method for estimating the initial values of all parameters in a model. To perform this estimation process, the administrator should set up an independent event within the ModelMaker file (see the separate documentation on creating models). On this screen, the name of the independent event should be specified. This event will be triggered at the start of the analysis when using the model. This list

shows all independent events contained in the model. The administrator should choose the correct event from the list. This list is *not available* for the Sediment Partitioning Study type, for which initial parameter estimation is not required.

- DTX Calculation

  The degradation time (DT) values are calculated after ModelManager has optimized a model against experimental data. To perform this calculation, the administrator should set up an event within the ModelMaker file (see the separate documentation on creating models). The optimized parameter values are used in this calculation. The list in the Wizard screen shows all events in the model specified. The radio buttons to the right of the list allow the administrator to switch between lists of independent and component events in the model. *Independent* events should be specified for DT calculation for all study types except Sediment Partitioning, where the nature of the DT calculations make *component* events more appropriate. At the end of the model analysis, independent events are evaluated, whereas component events are activated and a model run is started. See the separate document on the configuration procedure for ModelMaker models, which contains more detail on the above.

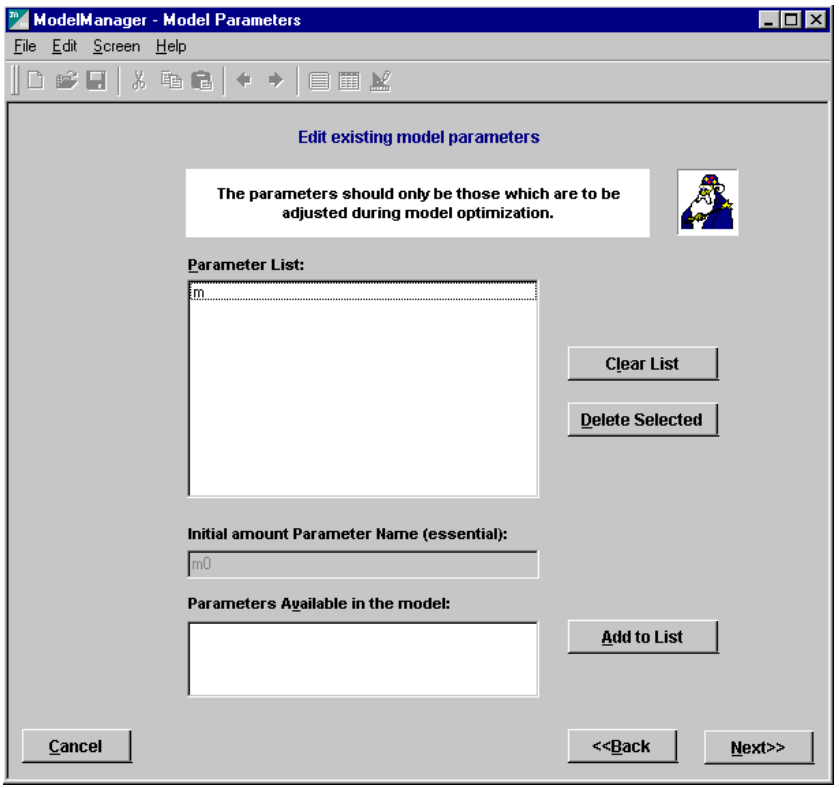
- Clear Selection

  This clears the current selection of the list of events under the heading “DTX Calculation” . This is included so that any event selection may be cleared if the administrator changes their mind and wishes not to specify an event.

Model Parameters

The Model Parameters screen enables the administrator to specify which model parameters are to be exposed to the end user, i.e. which parameters should be available to be optimized during ModelManager analyses.

The Model Parameters screen



The screen contains the following controls:

- Parameter List — This is a list of parameters currently selected for inclusion in ModelManager analyses for this model. If the user is currently editing a new model, this list will initially be empty.

- Initial Amount Parameter Name (essential) — ModelManager (EK) assumes models contain a parameter specifying an initial amount for all study types (except for Parent with Multiple Applications). This box displays the default name of this initial amount parameter, whose name is fixed as “M0”.

- Parameter Name — This box contains the name of the currently selected parameter in the list of available model parameters (see item below).

- Parameters Available in the model — This list contains all parameters that are in the model specified by the user. If the model contains no parameters this list will be empty. It is to remind the administrator which parameters are available to them.
- Add to List Button — When the user clicks on a parameter in the "Parameters Available in the model" list, its name will appear in the Parameter Name box. To add this parameter to the list to be available to ModelManager, the "Add to List" button should be used. The administrator should not reenter the "M0" parameter as this is automatically included. Parameters may not be added to the list more than once.
- Clear List — This will delete the current entries in the Model Parameter List. This button provides a means of "starting again".
- Delete selected — The currently selected parameter in the list will be deleted when this button is clicked. The remaining parameters will be reordered.

Note that the list of these parameters on the parameter configuration screen in ModelManager, when viewed by the end user, will appear in the *same order* as they are specified on this Model Parameter screen.

Note also that when the administrator is editing models for the Parent with Metabolite study types, they will be presented with an extra control at the top of the screen which is used to identify the component that are associated with the model parameters. To add parameters to the parent, select "parent" from the drop down list; to add parameters to the first metabolite, select "metabolite1", and so on.

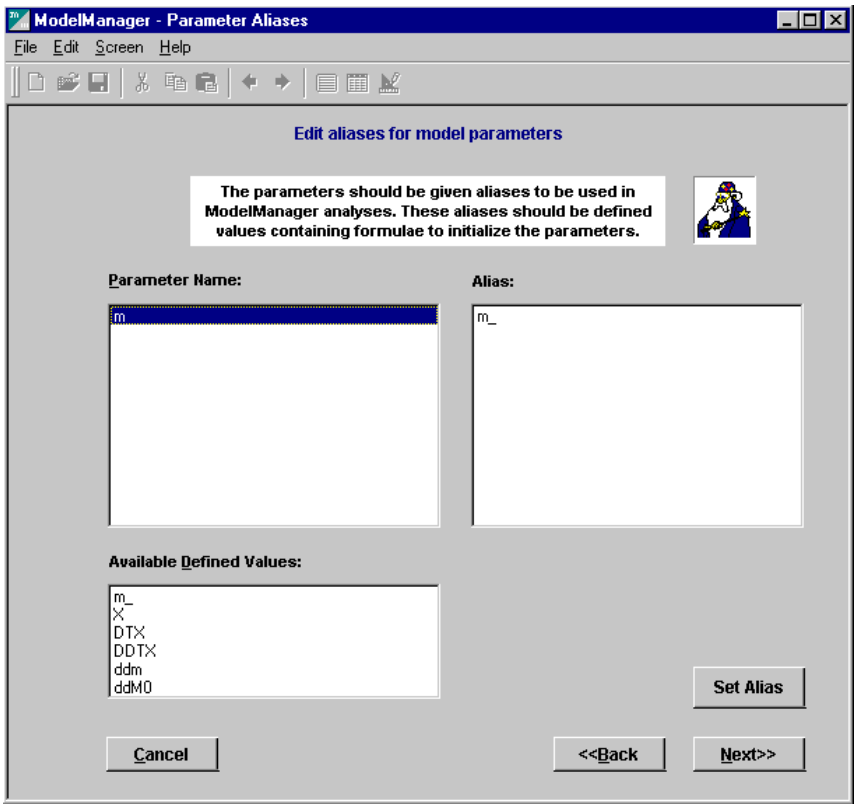
Parameter Aliases

The method of parameter initialization implemented in ModelManager is done by associating each parameter to be initialized with a component in the ModelMaker model. This component contains a formula that will calculate an initial estimate of that parameter. We refer to this component as an "alias" for that

parameter. See the separate documentation on the procedure for creating ModelMaker files for use in ModelManager (EK). These aliases are in fact *defined value* components in the ModelMaker file. Before optimization, ModelManager enters DT estimates into the ModelMaker model, from which the aliases calculate values using their formulae. ModelManager then retrieves the values of the aliases and uses them for the initial parameter values.

Note that this screen will not be shown by the wizard if the administrator is editing models in the Sediment Partitioning study type, as initial parameter estimation is not implemented for this study type.

The Parameter Alias wizard screen



This screen contains the following controls:

- Parameter Name List — This is a list of all the parameters selected by the user from the previous screen. An alias should be

Name List specified for each of these entries.

- Parameter Alias box — This box is where the administrator should enter the name of the alias to be associated with the currently selected parameter in the list.

- Defined Values list — This list shows all the defined values contained in the model currently being edited.

The administrator should go through the following process:

1. Select a parameter from the main list labeled “Parameter Name”.

2. Select a parameter alias from the list labeled “Defined values Available in the model”. The selected alias will appear in the text box labeled “Parameter Alias”. The parameter will then be associated with this alias when exiting the screen, or when the administrator selects another parameter from the main list.

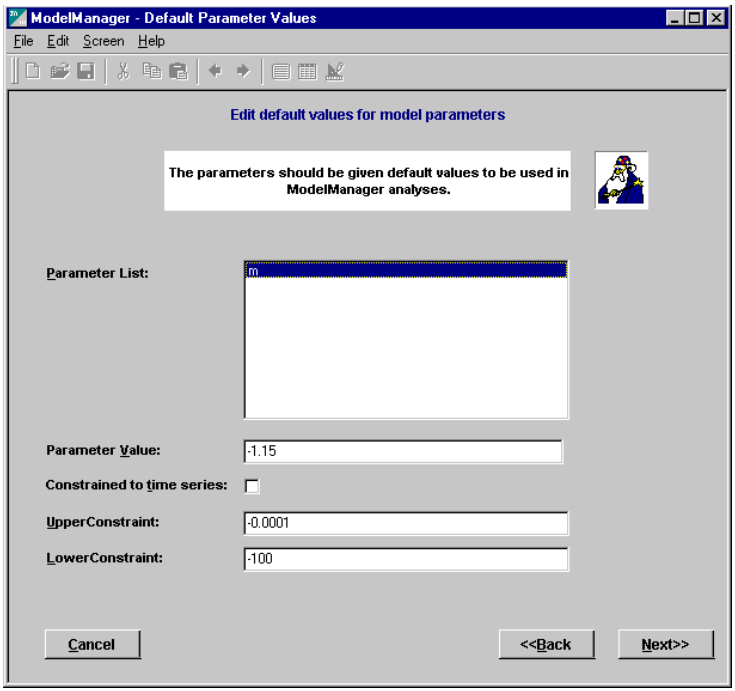
3. Select another parameter from the main list labeled “Parameter Name” and follow step 2 above. Repeat this process until aliases have been associated with all parameters in the main list. The administrator may then proceed to the next screen.

Note also that when the administrator is editing models for the Parent with Metabolite study types, they will be presented with another drop down list similar to the model parameters screen. As before, to specify parameter aliases for the parent parameters, select “parent”; to specify parameter aliases for the first metabolite, select “metabolite1”, and so on.

Default Parameter Values

This screen enables the administrator to enter default values and constraint ranges for all parameters included by the administrator for this model.

The Default Parameter Values screen



The screen contains the following controls:

Parameter List	This is a list of all parameters included in this model.
Parameter Value box	This control is for the administrator to specify the default initial value for the parameter currently selected in the parameter list.
Upper Constraint box	This control is to specify the default upper bound on the parameter values during optimization.
Lower Constraint box	This control is to specify the default lower bound on the parameter values during optimization.
"Constrain the parameter to the time series" check box	Some parameters, such as the breakpoint in the Hockey Stick model, must be constrained to the length of the time course data entered. Values outside of this time range are not sensible. Setting this check box will constrain this parameter to the time range of any dataset

to which this model is optimized. Setting this check box will disable the upper and lower constraint boxes for this parameter as they are no longer relevant.

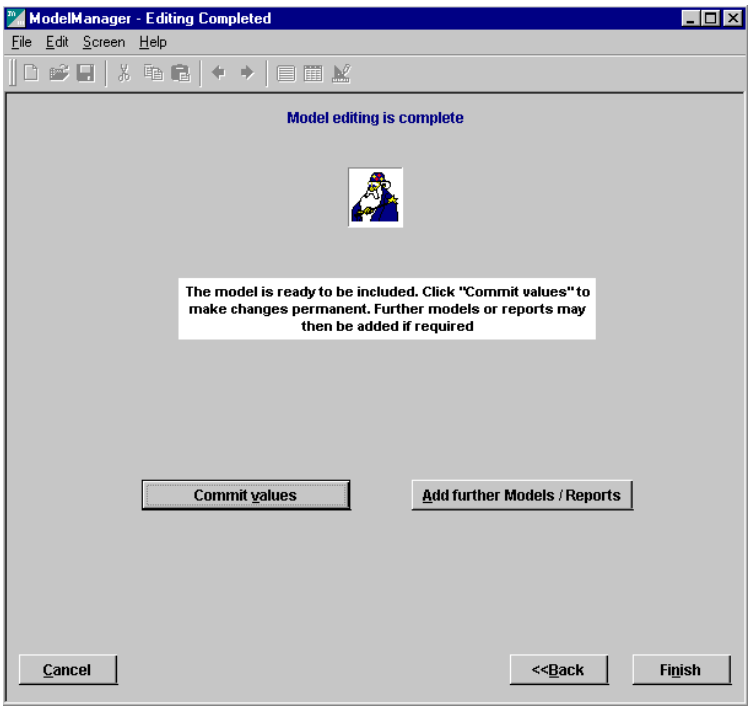
Once the administrator has completed the default values for all parameters in the list they will be allowed to proceed to the next screen.

Note also that when the administrator is editing models for the Parent with Metabolite study types, they will be presented with an extra drop down list box, similar to the control added to the model paramters and model aliases screens. In the same way, to specify parent parameter default values, select “parent” from the list box; to specify default values for the first metabolite, select “metabolite1”, and so on.

Wizard Final Screen

Once the administrator has completed all fields satisfactorily in all screens of the Wizard, they will be presented with the final screen:

The final screen is the Editing Completed screen



The title at the top of the screen will say either “Model editing is complete” or “Report editing is complete” depending on what the administrator chose to edit.

The screen contains the following controls:

- Commit Values button — Clicking this button commits the changes the administrator has made using the wizard.
- Add further Models / Reports — This button will take the administrator back to the opening screen of the Wizard, asking them whether they wish to commit any changes that have not yet been committed with the prompt “Do you want to save your changes?”
- Cancel button — This button will leave the wizard, asking the administrator whether they wish to commit any changes that have not yet been committed with the prompt “Are you sure you want to exit the Wizard and discard your changes?”

- Back button — This button allows the administrator to go back into the wizard and refine or correct any changes they have made.
- Finish button — This button returns the administrator to ModelManager, asking them whether they wish to commit any changes that have not yet been committed with the prompt “Do you want to save your changes?”.

If the administrator chooses to commit any changes they have made using the Wizard, they will be notified if the changes have been committed successfully with the Database updated message box. The changes have then been committed to the configuration databases. The changes made will be included in all new analyses that are constructed by the user.

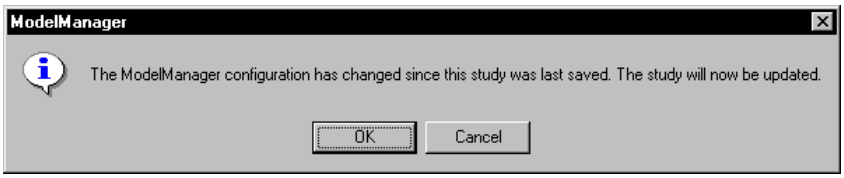
Confirmation dialog on successful update of configuration databases

Alterations to the reports are handled completely transparently to the user. There is no impact on the results contained in an existing study. The user will merely see a different list of available reports when they enter the analysis configuration screen whether they have opened an existing study or a new study.

Updated models are available to both new and existing studies opened by the user.

When the user opens an existing study that was created with an earlier model configuration of ModelManager, they will be presented with the message box below

Information dialog issued before existing studies are updated to new configuration.



The user may select OK to proceed and have the study updated. The update will not be committed to disk until the user saves the file manually. The user could therefore open the existing study and examine the changes, and then reject the update by closing the study without saving it. They may also choose "Cancel" from the dialog shown above to prevent the study being updated at all. This will return the user to the opening screen of ModelManager.

The administrator should note the following points about the procedure by which ModelManager will update the models in an existing study:

- Every effort has been made to ensure that existing results are preserved when updating a study. By results we refer to optimized parameter values. However, the user will have to run analyses again in order to generate reports showing model fits etc.

- If a model has been removed from the configuration using the Wizard, then this model will *not be available* in the updated study, and so any results contained in the study that involve this model will be *removed*.
  The remaining models in the study will be re-indexed, e.g. Say we have models with indices 1, 2, 3 and 4 in a study. If model 3 is removed from the configuration using the wizard, then during the update of the study, all results of model 3 will be removed. The remaining models, 1, 2 and 4 will then be re-indexed as 1, 2 and 3. This procedure will be transparent to the end user, but should be noted by the administrator.

- If a model has been *changed* using the wizard, then results in existing studies may be retained when the study is updated. There are only three changes to a model that will cause results to be lost: Alterations to the *number* or *names* of parameters available in a model, or if the model *name* (not filename) is changed.

Changes to the model filenames, abbreviation names, bitmaps, parameter aliases, initial parameter estimation and DTX calculation events or parameter default values are all changes that may be made to a model with *no impact upon the results* contained in existing studies. The only side-effect is that the user must perform a new analysis before reports may be generated.

ModelXchange

ModelXchange is a systems administration function that allows users to export and import a set of models. When a ModelXchange file is imported, the models and bitmaps are copied to the appropriate directories and the parameter information (e.g. default values, aliases, etc.) is added to the configuration database in one step so that the user can perform an analysis with the new models immediately. This allows a model set to be distributed quickly and easily without any prior knowledge being required about it, e.g. which parameters go with which compartments, the constraints on the parameters, etc. In effect, ModelXchange has all the wizard functionality wrapped up into one menu option.

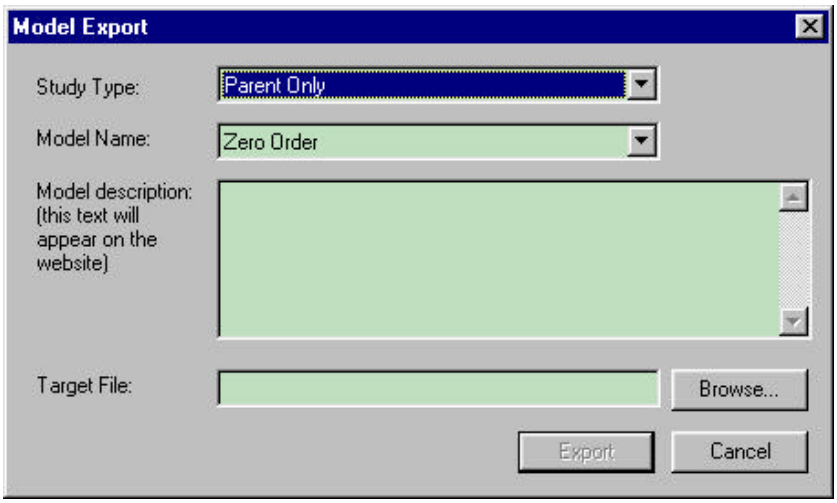
Cherwell Scientific provides a web site www.modelxchange.com for exchanging these files with other users of ModelManager.

Exporting Models

Before a model set can be exported with ModelXchange, it needs to be added to ModelManager and configured with the wizard as normal. When this has been done, ModelManager knows enough about the models to create the ModelXchange file.

ModelXchange is only available when the application is in systems administration mode, i.e. the user has entered the supervisor password. The menu option is only available from the start screen (where the user chooses to create a new study, open an existing study, or view the results of an existing study). The following screen is presented when the user selects File->ModelXchange->Export.

Model Export Dialog

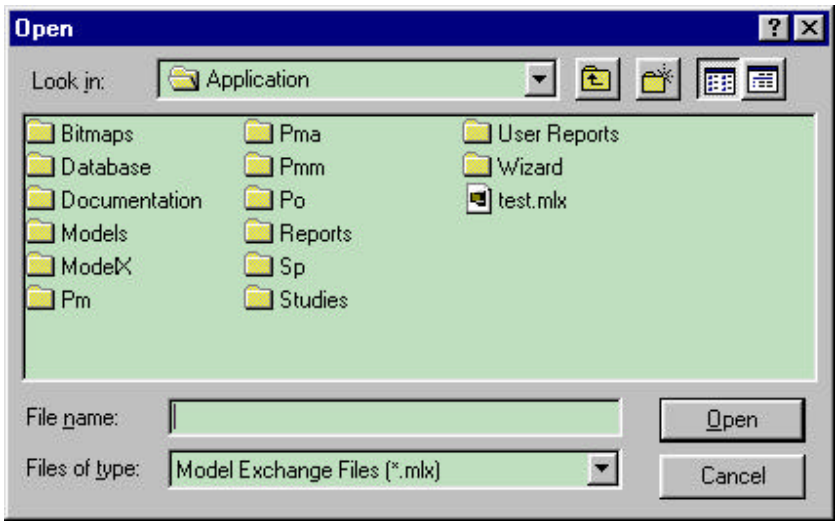


The study type drop down list box contains all the studies currently available in ModelManager. When the selection is changed, the Model Name list box is updated to include all the named model sets available in the analysis configuration screen for that study type. The study type / model name combination tell ModelManager which model set is going to be exported.

The Model description field allows the user to enter some extra information about the model. This is a mandatory free format text box with a limit of 1000 characters. If the ModelXchange file is to be added to the web site, this description will be used by other users to determine whether the model is of interest to them, so it is important that this field is completed. The export button will not be enabled until the text box contains some text.

Once the user is satisfied with the information on the export dialog, he needs to specify the name of the files that will be created. When the Browse button is pressed, the following dialog is displayed:

ModelXchange export browse dialog



This dialog is used to specify the name and location of the target file. Select the directory where the ModelXchange file should be saved, and enter the filename. It is not necessary to add the .mlx extension. Alternatively, an existing file can be overwritten by selecting it. Pressing the “open” button copies writes the full target filename (including path) to the export dialog.

Finally, to perform the export, press the “Export” button. If the export was successful, the following dialog is shown:

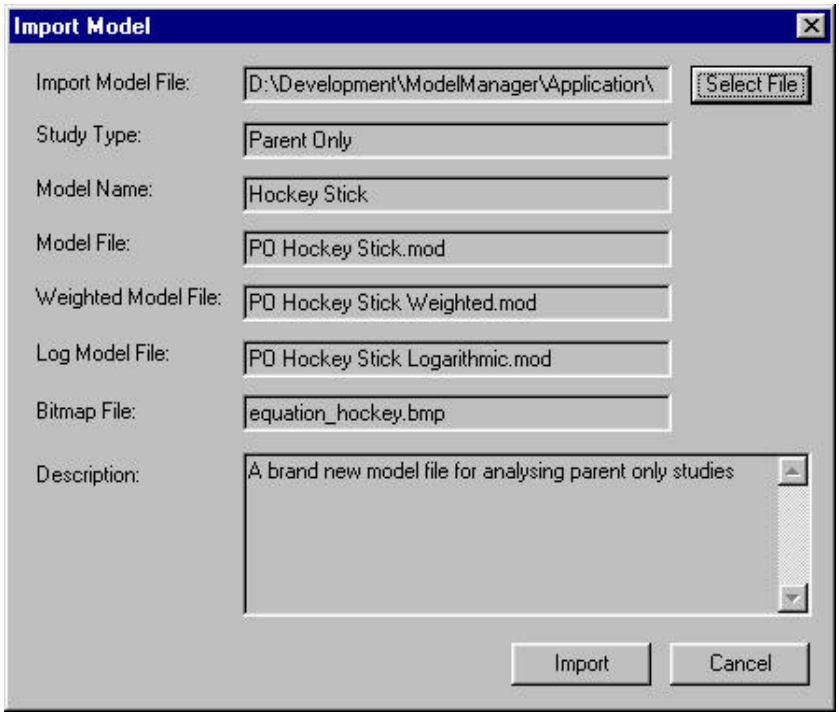
Successful model export dialog

This process creates two files. In this case, the test.mlx file is the ModelXchange file. Another file, called test.txt is created in the same directory as the .mlx file. This contains a textual description of the model set just created, e.g. the study type, the model name, and the comments added to the exchange file. This file can be edited to contain any other important information. Both of these files should be uploaded to the web site to give other users information about the ModelXchange file.

Importing Models

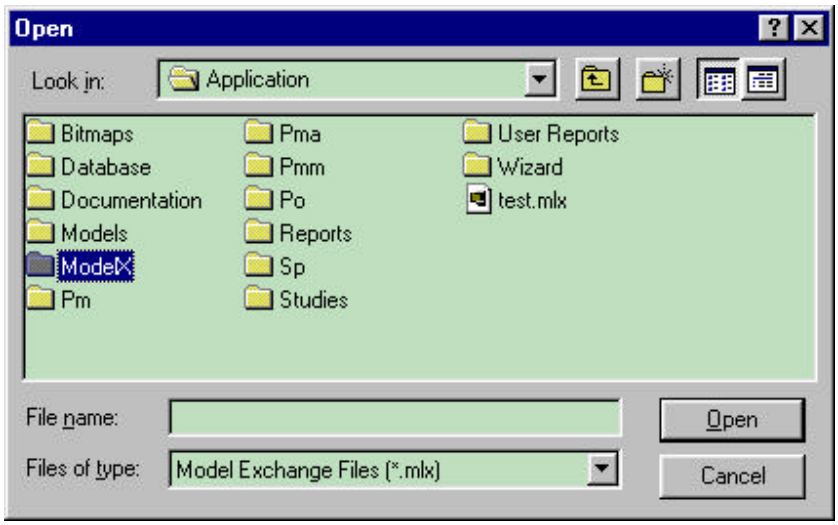
Once a ModelXchange file has been created, it can be imported by selecting File->ModelXchange->Import. Again, this option is only available from the starting screen. When this option is selected, the following dialog is presented to the user:

Import Model dialog



The ModelXchange file to import is selected by pressing the “Select file” button. This presents the following dialog to the user:

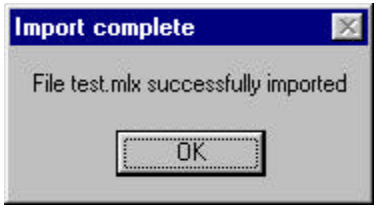
Select ModelXchange file



The exchange file to import is chosen by selecting the appropriate file and pressing "open" or by double clicking on the file. Once the file is selected, the dialog is closed and the information is retrieved from the .mlx file to populate the fields on the Import Model dialog. The user can see the names of the model files that are included in the model set, should he want to examine the file with ModelMaker. The description of the model set is also completed.

To import this file, press the "Import" button. If the import is successful, the following dialog is shown:

Successful Model import



The new model set is then available to be selected in the analysis configuration screen for that study type.

8. ModelMaker Models

This chapter presents an introduction to the assumptions that ModelManager (EK) makes about the ModelMaker models that it uses during analysis.

We do not provide a comprehensive description of the available functionality and component types contained within ModelMaker. For this information, the reader is referred to the ModelMaker online help facility.

Initial Parameter Estimation

Parameters are those components adjusted by ModelMaker when fitting a model to experimental data. ModelManager provides an interface to this process. The parameters in the ModelMaker models are available to ModelManager so that the user may select which parameters should be included in the optimization process, and what their initial values are and what constraints should be placed on their values.

The ModelManager user enters experimental data measurements of the degradation of a compound to which they wish to fit a model. They may also choose to enter estimates of how quickly the compound is degrading. The user then may choose to initialize parameter values in ModelManager according to either the experimental data entered, or their degradation estimates.

Functionality has been built into the ModelMaker models in order to perform initial parameter estimation. The model contains *defined values* to which ModelManager writes the estimates that were entered by the user. Other defined values in the model contain formulae from which the parameter values should be calculated (these formulae use the degradation estimates to calculate their estimates). If the user has entered no or insufficient estimates, then ModelManager automatically calls the *independent event* of the model, which works out its own estimates of the degradation from the experimental data.

The components to perform this process are the only necessary components for ModelMaker models to be available to ModelManager.

M0 Parameter

The parameter m0 (not case sensitive) represents the initial amount of parent compound in the system. ModelManager assumes its existence in the ModelMaker models for all study types except Parent with Multiple Applications.

DT Calculation

As mentioned above, for ModelManager to perform initial parameter estimation it is necessary for the ModelMaker models to contain a number of components. If calculation of DT (degradation time) values using the optimized parameter values is also required, then the administrator must also specify a separate event in the model which is to be executed at the end of model optimization.

The following components should be specified in order for the DT calculation to be performed, and the values read from the model by ModelManager:

Defined value components are required to hold:

1. The degradation percentage at which to calculate the DT value. The values calculated by ModelManager are DT50 and DT90 plus DTX where the X refers to the value entered by the end user on the Analysis Configuration screen. So the values 50, 90 and X are written in turn to this defined value by ModelManager.

2. The DT value itself.

3. The error on the DT value.

   In addition the superuser should specify and *event* to calculate the DT value from the optimized parameters. This is discussed in the following section.

   Parent Only, Parent with Multiple Applications and Parent and Metabolite study types “re-use” component used for the initial parameter estimation for the first two items in the list, and only

have an extra component for the DT error value (typically named DDTX).

Sediment Partitioning uses a different method of DT calculation than the other study types because there is no analytical method of calculating the DT in the water and the overall DT.

ModelManager assumes these components are included in the ModelMaker models it uses. If any of these components is absent from a model, ModelManager should still perform correctly, but some calculations may not be completed.

Below we show the relevant defined value names that should be used for each study type. These names are hard-coded into the visual basic scripting language behind the Analysis Configuration screen for each study type[6].

Not all models will require all components in order to perform its calculations, for example only the Hockey Stick model of the Parent Only study type requires the estimate of the breakpoint time (see entry "BT" in section 0). For each model in the Parent Only study type, ModelManager goes through the list contained in section 0 and checks to see if the model contains a defined value of this name. Only if this is present will ModelManager write the relevant value to this component.

Note that the error values on the DT values calculated in the ModelMaker model and written to the block DDTX should be *standard errors*. These are later converted to the 95% confidence limits that are quoted in the ModelManager database. This calculation is done in the visual basic code behind the Analysis Configuration screen, according to the student's t-distribution.

[6] Contained in the "EventClick" event of the "Go" button of this screen. The Administrator may change these hard-coded names if they wish. See the separate documentation on the ModelManager GUI for more details on events and the scripting language behind ModelManager screens.

The names and number of the defined value components that are assumed to be present in ModelMaker models varies between study types as follows:

Parent Only

Name	Purpose
X	Percentage for the first user DT estimate during initial parameter estimation. Percentage for DT value during DT calculation.
DTX	First user DT value estimate during initial parameter estimation. Calculated DT value during DT calculation.
Y	Percentage for the second user DT estimate during initial parameter estimation.
DTY	Second user DT value estimate during initial parameter estimation.
BT	User estimated break-point time during initial parameter estimation.
DDTX	Calculated standard error on the DT value during DT calculation.

Parent with Multiple Applications

Name	Purpose
X	Percentage for the first user DT estimate during initial parameter estimation. Percentage for DT value during DT calculation.
DTX	First user DT value estimate during initial parameter estimation. Calculated DT value during DT calculation.
Y	Percentage for the second user DT estimate during initial parameter estimation.
DTY	Second user DT value estimate during initial parameter estimation.

		parameter estimation.
	DDTX	Calculated standard error on the DT value during DT calculation.

Parent and Metabolite

Name	Purpose
PX	Percentage for the first user Parent DT estimate during initial parameter estimation. Percentage for Parent DT value during DT calculation.
PDTX	First user Parent DT value estimate during initial parameter estimation. Calculated Parent DT value during DT calculation.
PY	Percentage for the second user Parent DT estimate during initial parameter estimation.
PDTY	Second user Parent DT value estimate during initial parameter estimation.
MX	Percentage for the first user Metabolite DT estimate during initial parameter estimation. Percentage for Metabolite DT value during DT calculation.
MDTX	First user Metabolite DT value estimate during initial parameter estimation. Calculated Metabolite DT value during DT calculation.
MY	Percentage for the second user Metabolite DT estimate during initial parameter estimation.
MDTY	Second user Metabolite DT value estimate during initial parameter estimation.
DPDTX	Calculated standard error on the Parent DT value during DT calculation.

	value during DT calculation.
DMDTX	Calculated standard error on the Metabolite DT value during DT calculation.

Sediment Partitioning

Name	Purpose
X	Percentage for DT value during DT calculation.
DTsed	Calculated Sediment DT value during DT calculation.
DTwc	Calculated Water Column DT value during DT calculation.
DTtot	Calculated Overall DT value during DT calculation.
DDTsed	Calculated standard error on the Sediment DT value during DT calculation.

Events

ModelManager uses two events during the initial parameter estimation and DT value calculations as follows:

- Initial Parameter Estimation

  If any of the user DT estimates required by the model for the initial parameter estimation calculation are not specified, this event is triggered in order to estimate DT values from the experimental data. This Event is usually called “DATA” though may be named “DATA1” in Parent with Multiple Applications models.

- DT Calculation

Once the optimization process is complete ModelManager triggers an event to calculate the DT values and their errors from the optimized values. This Event is usually called “CALC”. The event uses the value that it finds in the defined value that holds the degradation percentage and writes the DT value and error that it calculates (using this percentage and the optimized parameter values) to the defined values that hold the DT value and its error. This process is repeated using percentage degradation values of 50 and 90 (by default) and the user-specified value X as entered on the Analysis Configuration screen (see separate documentation on the ModelManager GUI).

Variable Names

ModelManager makes a few assumptions about the names of variables contained in the models that it is to use for analysis.

The administrator should set the model up with

- Components with suitable names (see later)
- A Model Data table should be formed and associated with the components of the model

For example, the Parent Only study type models the degradation of a single compound with time. Refer to the model PO Zero Order.mod, contained in the folder “Models” in the installation directory of ModelManager.

This model contains a variable with name PARENT which describes the degradation of the parent from an initial value. The model file also contains a tab named “Model Data” which contains a blank table with table headings “t” and “PARENT”. These columns are associated with the parameter “t” and the variable PARENT respectively. Double-click the heading of each column to view the definition of that column.

When the Model is optimized, model parameters are adjusted to fit the variable PARENT with the data in the Model Data columns.

Note that names of components and table columns are not case-sensitive in ModelMaker or ModelManager.

In all model files included in the Parent Only study type, the component whose value describes the degradation of the initial compound should be denoted “PARENT”.

The assumed names of the components that are assumed in ModelMaker models varies between study types as follows (note that none of the following names are case sensitive):

All Study Types

The name of the independent variable should be **t**.

Parent Only

Component Name	Purpose
PARENT	Component whose value describes the degradation of the Parent compound over time.

Parent with Multiple Applications

Component Name	Purpose
PARENT	Component whose value describes the degradation of the Parent compound over time.
DATA	Lookup table containing application data to be used to specify the amounts applied as a function of time. See the model file e.g. Pma Zero Order.mod for an example.

Parent and Metabolite

Name	Purpose
PARENT	Component whose value describes the degradation of the Parent compound over time.
METABOLITE	Component whose value describes the degradation of the Metabolite compound

over time.

Sediment Partitioning

Name	Purpose
MWC	Component whose value describes the degradation of the Water Column concentration over time.
MSED	Component whose value describes the degradation of the Sediment concentration over time.

Weighting Models

ModelManager allows the weighting of model data during the parameter estimation process. To allow this to happen two different ModelMaker model files should specified. One of these files should be for un-weighted fitting and the other for weighted fitting. These two models should be exactly the same, except for their Optimization configuration as follows:

- Un-weighted models

  In ModelMaker, the “Least squares” check box on the “Definition” Tab of the Optimization dialog box must be checked. The Optimization dialog box may be viewed in ModelMaker by selecting “Optimize” from the “Model” menu.

- Weighted models

  In ModelMaker, the Least squares check box on the “Definition” Tab of the Optimization dialog box should not be checked. On the “Advanced” tab of the dialog box, the “mean observed” entry should be selected from the radio buttons under the heading “Data Weighting”. The Optimization dialog box may be viewed in ModelMaker by selecting “Optimize” from the “Model” menu.

Password Protection

ModelMaker now allows simple password protection to be applied to models: Adding a password to a ModelMaker model does not affect its operation with the ModelMaker Calculation Engine.

Current Password

The password currently used to protect the model files is:

```
snell
```

Adding Password Protection

To add a password to a ModelMaker model select Password Protection from the File menu. This invokes the Add Password Protection dialog.

The Add Password protection dialog

Enter the desired password in the Password and Verify fields and press OK.

When the model is opened, the user will be asked to enter the valid password. If an invalid password is entered, the user will not be able to open the file.

Changing a Password

To change a password of a ModelMaker model open the model and again select Password Protection from the File menu. Enter the new password in the Password and Verify fields and press OK.

Removing Password Protection

To remove password protection from a ModelMaker model open the model and select Password Protection from the File menu.

Delete the contents of the Password and Verify fields and press OK.

When the model is opened, the user will no longer be prompted for a password.

Model Bitmaps

When constructing a new model, the Administrator may wish to create a bitmap of the model equation for inclusion on the Parameter Configuration screen of ModelManager. The process of including a bitmap is described in the separate documentation on the ModelManager Wizard. The Administrator is free to create bitmaps for the new models (or change the existing ones) using any bitmap editor such as the Microsoft Paint accessory supplied with Windows. The bitmaps that are supplied with ModelManager were constructed using Microsoft Equation Editor (accessible through Microsoft Word) and then copied into Microsoft Paint.

Any number of colors may be used, but the Administrator should be warned that bitmaps containing a large number of colors may become very large in size. Microsoft Paint allows the bitmap to be saved with various numbers of colors. It is recommended that the background color of the bitmap be left white as this will appear transparent on the Parameter Configuration screen, and that the bitmap size is 340 x 130 pixels (width x height) or less.

9. ModelManager Databases

Introduction

The following list of databases are associated with ModelManager:

- Clean Study.mdb

  This database contains the minimal information required to construct a new study. When the user creates a new study, this database is copied to the filename mman.mdb in the installation directory and then accessed by ModelManager.

- Config.mdb

  This database contains read-only information about all units, models and report names available to ModelManager. Information may be added and removed from this database, possibly using the wizard, and this information will always be included in new and existing studies opened by ModelManager.

- Mman.mdb

  This database is the "working" database used by ModelManager. New studies are created by the file Clean Study.mdb being copied to this file name. When opening existing studies, the appropriate study database is copied to this filename. If the user wishes to save a particular study, the file mman.mdb is copied to the appropriate filename for the saved study.

- Wizard.mdb

  This database is the "working" database of the ModelManager Wizard facility. This contains information about new models and reports specified by the administrator. As the administrator enters information on the new model or report to be added, then this information is stored in this database while they inside the Wizard. This information is copied to the files Clean Study.mdb or Config.mdb as appropriate if the user wishes to save their changes at the end of the wizard.

Note that all saved studies have a database associated with them. They also have a <studyname>.sty file associated with them. When opening an existing study, the user should open the .sty file from within ModelManager, which then proceeds to open the appropriate database, rather than attempt to open the database directly.

When the user generates a report from their analysis, this report fetches information solely from the "working" database mman.mdb. No information is fetched from clean study.mdb, config.mdb or wizard.mdb.

Note that many of the database fields contained in mman.mdb will not be of use to the end user in creating reports. For example, many of the indices in the database are used by ModelManager for screen navigation and report creation etc.

Password Protection

All of the databases used and created by ModelManager have Microsoft Access security implemented. Complete documentation on the operation of Microsoft Access security can be found in the on-line help within Access. This section describes ModelManager's use of this function.

Access Work Group

In order to implement security on the ModelManager databases an Access workgroup has been created to control the database users and the access permissions of each user.

Note that it is only necessary for the Administrator, or others who wish to *alter* the databases in ModelManager to join the workgroup. The normal user need not join the workgroup.

The default user definitions are as follows, as defined in the workgroup:

User name	Default Password	Group	Comment
ModelManager	model	Admins	Used by ModelManager to access all the

			databases. Full read, write and change permissions
Admin	a	Users	Default user with read-only permissions
User		Users	Non-password protected user with read-only permissions

The Administrator who wishes to make changes to existing and new tables in the ModelManager databases listed below must first join the workgroup (see Joining the ModelManager Workgroup) and then open the database with the username “ModelManager” as shown above.

Default Workgroup

We have mentioned that the normal user of ModelManager will not need to join the ModelManager workgroup. Here we present a brief explanation of how password protection is implemented for the normal user..

Microsoft Access allows a user to be a member of one workgroup at a time. By default, the user will be a member of the *default* system workgroup. In this workgroup, when the user starts up Access, they are in fact being automatically logged on under the username “Admin”. The default workgroup requires no password for this username, so the user is not prompted to enter one.

The permissions of the ModelManager databases have been set up so that users logged on as “Admin” (whether they are part of the ModelManager, default or other workgroup) may only open these databases in read-only mode. Hence when the user starts up Access and is logged in as “Admin” automatically as described above, they may not alter any records in the database. Also, they may not change the permissions on the database - for example they may not specify that for database mman.mdb, users logged in as “Admin” may have write and change permissions etc.

Only those who have joined the ModelManager workgroup and have logged onto Access using the username “ModelManager” may change the databases in Access, or change the permissions of the database.

Joining the ModelManager Workgroup

Joining the ModelManager workgroup is done by performing the following steps:

- Without Microsoft Access running start the Microsoft Access Work Group Administrator (Wrkgadm.exe). Under Office 95 this has the following default location:

  C:\MSOffice\Access\Wrkgadm.exe

  Under Office 97 this has the following default location:

  C:\Windows\System\Wrkgadm.exe

- Press the Join button and locate the ModelManager work group file (mman.mdw) This has the following default location:

  C:\Program Files\Cherwell Scientific\ModelManager (EK)\Mman.mdw

- Press OK and you will receive the following message:

- Start Microsoft Access. You will be asked to enter a valid user name and password. The valid default user names and passwords are show in the table above.

Changing the ModelManager Database Password

Changing the ModelManager database password requires that

- the password for the ModelManager user name in the ModelManager Microsoft Access work group is changed

and

- the password defined within ModelManager itself is changed to match

To change the password form the ModelManager user name in the ModelManager Microsoft Access work group perform the following steps:

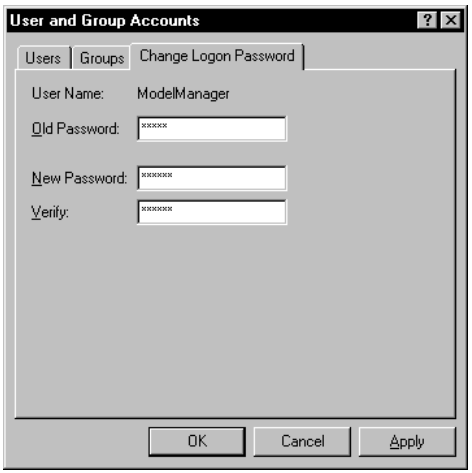
- Join the ModelManager work group as defined in the previous section.

- Start Microsoft Access using the ModelManager user name and password to give you full administrative rights to the database. You do not have to open a database.

- From the Tools menu select Security | User and Group Accounts… This invokes the User and group accounts dialog.

- On the Change Logon Password tab enter the new password and press OK

The Users and Group Accounts dialog



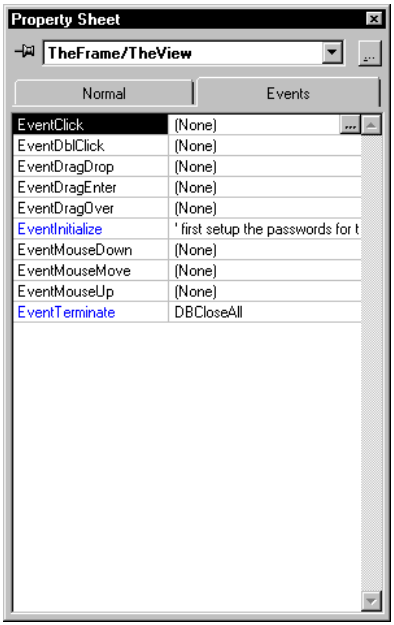
Note that if the Administrator removes the password for the username “Admin” by leaving the “New Password” and “Verify” fields blank on the dialog above, then no password dialog will appear then they subsequently open Access, as they will automatically be logged on as “Admin”. To force a login dialog to appear when opening Access, the Administrator should specify a password for the

Admin username. The password dialog will then appear whenever Access is opened, from which the Administrator may log in as either “ModelManager”, “Admin” or “User”.

To change the password defined within ModelManager itself to match the ModelManager Microsoft Access work group perform the following steps.

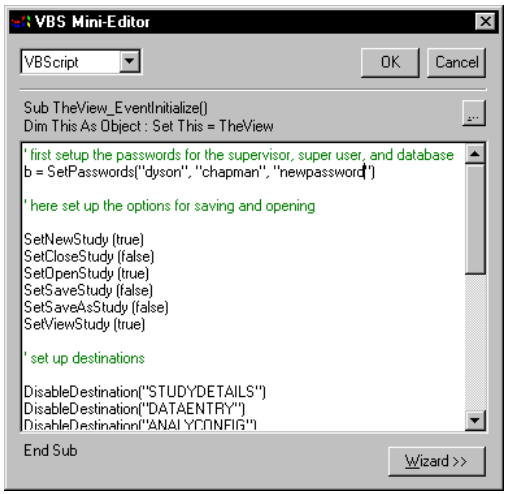
- Start ModelManager
- On the opening screen enter supervisor mode by selecting Supervisor mode from the Edit menu and pressing Ctrl-E
- Double-click the right button of the mouse on the screen background to access the properties of the screen

The Screen Property Sheet

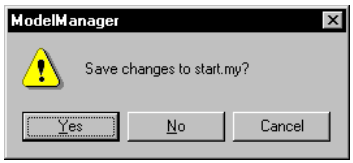


- On the Edit tab select EventInitialize
- In the script editor change the third parameter of the SetPasswords function to the new database password and press OK

The VBS Mini editor



- Press Ctrl-E and press Yes to save the changes



- Select User Mode from the Edit menu to enter user mode.

Re-joining the Default Workgroup

If the Administrator wishes to re-join the default Access workgroup, they should follow the steps described above, but specify the name of the default workgroup file, which has the following location under Office 95:

C:\MSOffice\Access\system.mdw

rather that the ModelManager workgroup file:

C:\Program Files\Cherwell Scientific\ModelManager (EK)\Mman.mdw.

Clean Study.mdb

The tables contained in this database are the same tables that are stored in the database for a saved study. The tables are listed in the following sections:

AnalyConfig

This table contains information about the configuration of the analysis specified by the user. It contains the following fields:

- AnalyFixedIntercept — Equals 1 if a fixed value of M0 was specified. Zero otherwise.
- AnalyEstimateMethodID — This specifies the index of the method selected for parameter configuration, e.g. grid search, user inputs etc. For example, in the Parent Only study type, 0 = Use existing parameter estimates, 1 = use the user inputs to calculate initial estimates and 2 = Perform parameter search to find initial estimates.
- AnalyLogTransform — Equals 1 if the data is to be weighted in the analysis. Equals 0 if the data is un-transformed. The field name is a legacy of an earlier version of ModelManager which log transformed the experimental data.
- AnalyReportID — This is an index number specifying the ID of the report selected in the analysis. There may be any number of reports available to the end user, and as these are added to the ModelManager configuration, they are given a zero-based index number 0, 1, 2 etc. which identifies each report.
- AnalyDTX — This is the value to calculate the DT value at (in addition to default calculations at 50 and 90).
- AnalyOptimiseType — This contains an index referring to the optimisation method selected. The current values are 0 for Marquardt

	and 1 for Simplex optimization.
• AnalyGenerateReport	Equals 1 if the report is to be generate in the analysis run. Equals 0 if no report is to be generated.
• AnalyAllowReport	Equals 1 if an analysis has been run for the current study, else has the value 0. For example, if the user creates a new study, enters data then opens the Analysis configuration screen, the button marked “Create”, which creates a report based on the current results, will be disabled. The create button remains disabled until some results have been created by running model(s) with the “Go” button.
• AnalyModelTotal	This specifies the total number of models *selected* in the analysis run.
• AnalyDatasetTotal	This specifies the total number of datasets *selected* in the analysis run.

Cases This table contains indices relating to the identities of all the datasets, models etc. for the current study run.

• caseID	Each model / dataset pair is assigned a Case ID. For example, for 2 models applied to 2 datasets, there would be 4 cases. CaseID in this case would run from 1 to 4.
• modelID	Each model in the database is assigned a model ID. This value is used by ModelManager to identify which model should be used. This is a one-indexed integer, so if there were four models in a Study Type these would be indexed 1, 2, 3 and 4.
• datasetID	Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset should

be used. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.

- inparamsID — Obsolete field.
- outparamsID — Obsolete field.
- fitID — This index is a replica of the caseID. It is used to identify the fit corresponding to the model/dataset pair.
- studyTypeID — This is an index which specifies the current study type. StudyTypeID is a zero-indexed value which is assigned as follows:

    0. Parent Only
    1. Parent with Multiple Applications
    2. Parent and Metabolite
    3. Sediment Partitioning
    4. Parent with two metabolites

DatasetInfo This table contains all the associated information relating to a particular dataset, apart from the time course data itself. Each dataset has an entry in this table, and the entries are indexed by the dataset ID number.

- DatasetName — This is the user-defined name of the dataset.
- DatasetNominal — This is the nominal applied amount, specified on the Experimental data entry screen.
- DatasetUserApplied — This is the User-defined applied amount, specified on the Experimental data entry screen.
- DatasetComments — This field contains the comments associated with the dataset, specified on the Experimental data entry screen.

• DatasetID	Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset should be used. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.
• DatasetParentEst1 Percent	This is the first estimate of the percentage of degradation of the parent, as specified on the Experimental data entry screen.
• DatasetParentEst1 Units	This is the amount of time over which the first estimate of parent degradation took place, as specified on the Experimental data entry screen.
• DatasetParentEst2 Percent	This is the second estimate of the percentage of degradation of the parent, as specified on the Experimental data entry screen.
• DatasetParentEst2 Units	This is the amount of time over which the second estimate of parent degradation took place, as specified on the Experimental data entry screen.
• DatasetMetabolite Est1Percent	This is the first estimate of the percentage of degradation of the metabolite, as specified on the Experimental data entry screen. This field is only used for Parent and Metabolite studies.
• DatasetMetabolite Est1Units	This is the amount of time over which the first estimate of metabolite degradation took place, as specified on the Experimental data entry screen. This field is only used for Parent and Metabolite studies.
• DatasetMetabolite Est2Percent	This is the second estimate of the percentage of degradation of the metabolite, as specified on the Experimental data entry screen. This field is only used for Parent and Metabolite studies.
• DatasetMetabolite Est2Units	This is the amount of time over which the second estimate of metabolite degradation took place, as specified on the Experimental

	data entry screen. This field is only used for Parent and Metabolite studies.
• DatasetBreakpoint	This field is used for the Hockey Stick model and specifies the estimated break point time in that model, as specified on the Experimental data entry screen.
• DatasetSelected	The value of this field is 1 if the dataset has been selected on the analysis configuration screen and 0 if the data is not selected for analysis.
• DatasetParentMetabolitePercent	This is an estimate of the percentage of parent degrading to metabolite, as specified on the Experimental data entry screen. This field is only used for Parent and Metabolite studies.

Datasets This table contains the time course data for each dataset, as specified by the user on the Experimental data entry screen. Each dataset is indexed by the dataset ID number.

• datasetID	Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset should be used. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.
• time	This column contains time values entered on the experimental data entry screen.
• measuredParent	This column contains measured Parent values entered on the experimental data entry screen.
• measuredMetabolite	This column contains measured Metabolite values entered on the experimental data entry screen.
• removedParent	This column contains measured Parent values entered on the experimental data

entry screen which have been removed from the analysis by placing an asterisk before the value in the data entry table.

- removedMetabolite — This column contains measured Metabolite values entered on the experimental data entry screen which have been removed from the analysis by placing an asterisk before the value in the data entry table.
- dummy — Obsolete field.

Degradation Estimates

This table contains the list of degradation estimates that the user entered for a Parent with two metabolites study.

- DatasetID — Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset should be used. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.
- ComponentId — Every component in the system being modeled in the system is given an id; for instance, the parent has an id of 1, the first metabolite has an id of 2, and the second metabolite's id is 3. This allows us to specify which component the estimate applies to.
- EstimateId — The estimate id stores the id of the estimates that the user supplied. In this way, it is possible for the user to provide as many estimates as necessary for a particular model. If 3 estimates were required of the percentage degradation on the parent, the estimates would have the ids 1, 2 and 3.
- PercentageDegraded — This field stores the percentage that the user entered as the estimate for the degradation of the component.
- TimeUnits — TimeUnits refers to the length of time that the component takes to degrade by the value stored in the PercentageDegraded field

DTValues This table contains the calculated DT value for each model fit to a dataset. The results from the fit of all dataset / model pairs are contained in this table, which is indexed by the case ID.

- CaseID — This is the case ID of the model / dataset pair for which the DT value is being reported.
- Name — This is either “DT50”, “DT90” or “DTX”, which denote whether the DT calculation is either of the default DT50 or DT90 calculations, or the user-defined DTX value.
- Value — This contains the calculated DT value from the analysis.
- ConfidenceHi — This contains the upper 95% confidence limit of the calculated DT value.
- ConfidenceLo — This contains the lower 95% confidence limit of the calculated DT value.
- type — This specifies whether the DT calculation refers to the parent (type = 0) or the metabolite (type = 1). For the Sediment Partitioning study this index refers to the water column (type = 0) or sediment (type = 1) DT calculation.

Fits This table contains the values of the model fits at of the requested output steps, as specified in the ModelMaker model. The number of output steps is typically 100. The results from all model fits are included here, and this table is indexed by the fitID. Note that results from sediment partitioning models are included in this table, with “water column” values in the “parent” column, and “sediment” values in the “metabolite” column.

- fitID — This index specifies the ID of the fit - i.e. the model / dataset combination.
- time — This column contains the time values of the fit.
- parent — This column contains the parent values of the fit.

- metabolite — This column contains the metabolite values of the fit.
- Metabolite2 — This column contains the values of the fit to metabolite 2
- Metabolite3 — This column contains the values of the fit to metabolite 3
- Metabolite4 — This column contains the values of the fit to metabolite 4
- Metabolite5 — This column contains the values of the fit to metabolite 5

InParams This table contains all the model parameters for all the datasets specified in a particular study. For example, if a study type has 4 associated models, and the user has entered 3 datasets, then there will be 12 combinations.

- caseID — This index specifies the case identifier for the model/dataset pair for the most recent analysis. If the model/dataset pair was not selected during the last analysis that was performed, this value of CaseID is 0.
- modelID — This index specifies the identifier of the model that the parameters refer to.
- datasetID — This index specifies the identifier of the dataset that the model parameters refer to.
- paramName — This field contains the name of the parameter.
- paramValue — This field contains the current value of the parameter, whether this is the initial estimate of the parameter or its optimized value.
- paramError — This field contains the current error on the parameter value contained in the field paramValue.
- confidenceHi — This field contains the upper constraint on the parameter value during optimization.

- confidenceLo — This field contains the lower constraint on the parameter value during optimization.
- fixed — This field has a value of 1 if the parameter is to be fixed in the optimisation. Its value is 0 if the parameter value is to be estimated during the analysis.
- initialEstimate — This field contains the type of initial estimate of the M0 parameter that was used in optimisation. 0 - Actual measured, 1 - Nominal applied, 2 - User-defined, 3 - Optimized value.
- lowerConstraint — This is the lower constraint that is to be put on the parameter value during optimization. The final optimized parameter value should not be smaller than this value.
- upperConstraint — This is the upper constraint that is to be put on the parameter value during optimization. The final optimized parameter value should not be greater than this value.
- type — This field specifies whether the parameter relates to the parent (= 0) or metabolite (=1) part of the model. This is used to specify which model parameters should be selected when choosing to optimize the parent then metabolite in the Parent and Metabolite study type.
- timeLike — This specifies whether the parameter is to be constrained to the range of the independent variable in the optimization, i.e. time. If this field is set to 1 then the parameter value may not fall outside the upper or lower time values of the time course data entered by the user, and the fields lowerContraint and upperConstraint are ignored. If this value is 0 then the lowerConstraint and upperConstraint fields are used to constrain this parameter.

Main This contains the current study type. It contains no other fields.

- StudyTypeID — This is an index which specifies the current study type. StudyTypeID is a zero-indexed value which is assigned as follows:

  0. Parent Only
  1. Parent with Multiple Applications
  2. Parent and Metabolite
  3. Sediment Partitioning
  4. Parent with two metabolites

Matrices This table contains the correlation and covariance values for the parameters estimated during the most recent analysis.

- caseID — This index specifies the model / dataset pair for which these correlation and covariance values were calculated.
- paramName1 — This contains the name of the first parameter for which the correlation and covariance values were calculated.
- paramName2 — This contains the name of the second parameter for which the correlation and covariance values were calculated.
- correlation — This contains the correlation matrix value for the given parameters.
- covariance — This contains the covariance matrix value for the given parameters.

MetaboliteFormation This table contains the estimates that the user supplies of the percentage of a component that forms another component, e.g. how much of the parent (component 1) forms metabolite 1 (component 2).

- DatasetId — Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset this estimate refers to. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.
- SourceCompId — This field identifies the id of the component that the estimate refers to. For example, if the estimate recorded was for how much of the parent forms metabolite1, the value in this field would be 1.
- ProducedCompId — This field identifies the component that is formed while a different component degrades. For instance, if the estimate recorded was for how much of the parent forms metabolite1, the value in this field would be 2.
- Percentage — This field records the actual estimate that the user supplied on the dialog. For instance, if the user supplied an estimate that 20% of the parent formed metabolite1, the value in this field would be 20.

ModelOptions This table contains the names of the integration and optimisation methods used in the study. This may be required for reporting purposes.

- RunIntegMethod — This contains the name of the integration scheme selected. Currently there are four possible values, “Euler”, “Mid-point”, “Runge-Kutta” or “Bulirsch-Stoer”.
- OptimMethod — This contains the name of the optimisation scheme selected. Currently there are two possible values, “Simplex” or “Marquardt”.

Models This table contains details of the models available in ModelManager when this study was last saved. The details may be required for reporting purposes.

- StudyTypeID — This is the number of the study type to which the model belongs.
- ModelID — This index refers to the identifier of the model.
- ModelName — This specifies the full model name. This name is the same for both linear and weighted models, e.g. “Parent Only”
- ModelAbbrevName — This specifies an abbreviated name for the model, which may be useful in report generation, e.g. “Parent Only” is abbreviated to “PO”.
- ModelUsed — This index specifies whether the model is selected (ModelUsed = 1) or not (ModelUsed = 0) in the list of models on the Analysis configuration screen.
- ModelBitmap — This field contains the filename of the model bitmap. This can be used to present the model equations in the reports.
- ExtraModelData — This field is used to store any extra data specific to a model. In the Parent with 2 metabolites study, this refers to the number of metabolites that the model can handle.

ModelStats This table contains details of the statistics of the model analysis.

- CaseID — This contains the index of the model / dataset pair for which these are the model statistics.
- Measure — This field specifies the type of statistic presented, e.g. “Total” refers to the total sum of squares, “Residual” to the residual sum of squares etc.
- DF — This field specifies the number of degrees of freedom.
- SumSquare — This field contains the sum of squares that was calculated for this particular model / dataset pair.

- MeanSquare — This field contains the mean square value that was calculated for this particular model / dataset pair.

MultipleApps

This table is only used for the Multiple Applications study type. It contains details of the list of applications.

- datasetID — This index refers to the dataset to which the application data belongs.
- time — This field contains the time values of the applications.
- actual — This field contains the Actual applied values, as specified on the Experimental data entry screen.
- nominal — This field contains the nominal applied values, as specified on the Experimental data entry screen.
- user — This field contains the user specified values, as specified on the Experimental data entry screen.

OptimOptions

This table contains entries for user-defined optimisation options. See the ModelMaker manual for more details on their mathematical meaning.

- OptimOverride — This field specifies whether the optimisation options should be taken from the optimisation configuration screen (OptimOverride = 1) or the default values in the model (OptimOverride = 0)
- OptimConvergeSteps — This field specifies the number of convergent steps after which the model is said to have converged satisfactorily.
- OptimInitLambda — This field specifies the initial lambda value to be used in Marquardt optimisation.
- OptimConvergeChange — This field specifies the convergence change to be used in the optimisation.

- OptimFractionChange — This field specifies the fractional change to be used in the optimisation.
- OptimMinChange — This field specifies the minimum change to be used in the optimisation.
- OptimRetryCount — This field specifies the maximum number of unsuccessful optimisation steps that are made before optimisation is said to fail.
- OptimMethodID — This field specifies the index of the optimisation method selected by the user. OptimMethodID is zero-indexed and has the value 0 for Marquardt and 1 for Simplex optimization.
- OptimGridSteps — This field specifies the number of grid steps to use when performing a grid search.

PMMData This table contains the time course data for a Parent with two metabolites study for each dataset, as specified by the user on the Experimental data entry screen. Each dataset is indexed by the dataset ID number.

- DataSetId — Each dataset in the database is assigned a dataset ID. This one-indexed value is used by ModelManager to identify which dataset should be used. For example if the user specified three datasets, then these would be indexed 1, 2 and 3.
- ComponentId — This field identifies the component that the measured data was entered for. This field is 1 indexed, where 1 indicates data for the parent, 2 stands for metabolite1 and 3 identifies metabolite3.
- time — This column contains time values entered on the experimental data entry screen.
- Measured — This column contains measured values entered on the experimental data entry screen for the component identified in the ComponentId field.

- Removed — This column contains measured values entered on the experimental data entry screen for the component identified in the componentId field, which have been removed from the analysis by placing an asterisk before the value in the data entry table.

Points This table contains the values of the model fits that were calculated, that correspond to the same time values as the entered dataset. The results from all model fits are included here, and this table is indexed by the fit ID. Note that results from sediment partitioning models are included in this table, with “water column” values in the “parent” column, and “sediment” values in the “metabolite” column.

- fitID — This index specifies the ID of the fit - i.e. the model / dataset combination.
- time — This column contains the time values of the fit. These values are the same time values as those specified in the dataset.
- parent — This column contains the parent values of the fit.
- metabolite — This column contains the metabolite values of the fit.
- Metabolite2 — This column contains the metabolite2 values of the fit.
- Metabolite3 — This column contains the metabolite3 values of the fit.
- Metabolite4 — This column contains the metabolite4 values of the fit.
- Metabolite5 — This column contains the metabolite5 values of the fit.

RunOptions This table contains entries for user-defined run options. See the ModelMaker manual for more details on their mathematical meaning.

• RunOverride	This field specifies whether the run options should be taken from the run configuration screen (RunOverride = 1) or the default values in the model (RunOverride = 0)
• RunAccuracy	This specifies the required accuracy of the integration method.
• RunMinVal	This specifies the minimum value to be tolerated in the integration.
• RunStartSteps	This specifies the number of starting steps to use in the integration.
• RunErrorScalingID	This index specifies the type of error scaling selected by the user. The options are proportional to the function (0), the function and its derivative (1) or to a constant value (2).
• RunValue	This contains the constant value to be used in the error scaling, should this option be selected by the user.
• RunIntegMethodID	This index specifies the integration method selected by the user. This has the value 0 for Euler, 1 for Mid-Point, 2 for Runge-Kutta (the default) and 3 for Bulirsch-Stoer integration.
• RunIntegAccuracy	This specifies the required accuracy of the model run-time integration function used in some of the Parent and Metabolite models.
• RunIntegSteps	This field is used to specify the maximum number of steps that are to be allowed in the model run-time integration function. This function is used in some of the Parent and Metabolite models. The maximum number of steps to be used is in fact 2 to the power of this value.

Screens

This table is used for correct operation of the ModelManager user interface through the various screens.

- StartID — Obsolete field.

Sediment

This table contains all the various parameters required for the Sediment Partitioning model. It is not used for any other study types.

- datasetID — This is the ID of the dataset to which the table entry refers.
- method — This index specifies the method of parameter constraint selected on the parameter configuration screen for sediment partitioning. 0 - None, 1 - User-defined, 2 - Calculated. Note that this is not the order in which these options appear on the Parameter Configuration screen of the Sediment Partitioning study type.
- Ksed — This field holds the K_{sed} value specified by the user on the parameter configuration screen for sediment partitioning.
- Kd — This field holds the K_d value specified by the user on the parameter configuration screen for sediment partitioning.
- pb — This field holds the ρ_b value specified by the user on the parameter configuration screen for sediment partitioning.
- theta — This field holds the θ value specified by the user on the parameter configuration screen for sediment partitioning.
- Usetheta — Obsolete field.
- OM — This field holds the OM value specified by the user on the parameter configuration screen for sediment partitioning.
- Vsed — This field holds the V_{sed} value specified by the user on the parameter configuration screen for sediment

partitioning.

- Vwc — This field holds the V_{wc} value specified by the user on the parameter configuration screen for sediment partitioning.

StudyDetails This table contains information about the study type, name, date etc.

- StudyNumber — This is the study number specified by the user on the Study Details screen.
- StudyName — This is the study name specified by the user on the Study Details screen.
- StudyDesc — This is the study description entered by the user on the Study Details screen.
- StudyDOA — This is the study date of analysis specified by the user on the Study Details screen. If a new study has been started then this value will be set to today's date. This field is not updated every time the study is opened.
- StudyUserName — This is the study user name specified by the user on the Study Details screen.
- StudyMeasureUnitsID — This index specifies the study measure units selected by the user on the Study Details screen. The index has the following values: 0 (ug/l), 1 (mg/kg), 2 (kg/ha), 3 (% of applied) or -1 (user-defined value not selected from the list).
- StudyMeasureUnits — This field contains the name of the measure units selected by the user on the Study Details screen.
- StudyTimeUnitsID — This index specifies the study time units selected by the user on the Study Details screen. The index has the following values: 0 (Weeks), 1 (Days), 2 (Hours), 3 (Seconds) or -1 (user-defined value not selected from the list).

- StudyTimeUnits — This field contains the name of the time units selected by the user on the Study Details screen.
- StudyNumDatasets — This is the number of datasets specified by the user on the Study Details screen.
- StudyComments — This is the study comments specified by the user on the Study Details screen.
- StudyCurrentDataset — This index is used by ModelManager during screen navigation, and specifies the current dataset on display on the screens. For example, if the user has four datasets in a particular study, and selects dataset 2 before exiting the Experimental data entry screen (for example) then this index records the fact that dataset 2 was selected on this screen. Thus when the user re-opens the Experimental data entry screen, ModelManager knows to display dataset 2.

StudyTypes This table contains a list of all the currently available study types and their ID numbers.

- StudyTypeID — This index contains the study type index value.
- StudyType — This field contains the name of the study type.

Config.mdb

This database contains lists of characteristics that may wish to be changed for all currently existing studies, for example the list of reports available, list of integration methods available etc.

Debug This table contains a single field used during the editing of ModelManager screens.

- DebugWindow — This switch, when set to 1, causes a pause in the operation of ModelManager, immediately before the database is opened (for either the Wizard or the main application). This allows persons with Supervisor authority to edit the commands which open the database. If the value of this switch is 0 then the database is opened automatically with no pause.

IntegMethods This table contains a list of all integration methods available in ModelManager.

- IntegMethod

  This field contains the name of the integration method.

- IntegMethodID

  This field contains the ID number of the integration method.

MeasureUnits This table contains a list of all the measure units available in ModelManager.

- MeasureUnits — This field contains the name of the measure units.
- MeasureUnitsID — This field contains the ID number of the measure units.

ModelParams This table contains information about all parameters associated with all models available in ModelManager.

- modelID — This is the identifier of the model to which this parameter belongs.
- StudyTypeID — This is the identifier of the study type to which the model parameters belong.
- type — This specifies whether the parameter is M0 (type = 1), a Parent parameter (type = 1) or a Metabolite parameter (type = 2).

- paramID — This is the one-based index value of the parameter. If there are three parent parameters in a model then they will have paramID 1, 2 and 3 respectively. Note that if there are also three metabolite parameters, then they will also have indices 1, 2 and 3. In these cases, the index "type" is used to separate parent and metabolite parameters.
- paramName — This is the name of the parameter.
- value — This is the initial (default) value of the parameter when used for the first time (e.g. in a new study).
- lowerConsraint — This is the initial (default) lower constraint to be placed on the parameter value during optimisation.
- upperConstraint — This is the initial (default) upper constraint to be placed on the parameter value during optimisation.
- paramCalcName — This is the name of the component in the model which is used to calculate the initial parameter estimate at the start of the optimisation.
- timeLike — This specifies whether the parameter value should be constrained by the limits of the time series of the experimental data. If timeLike = 0, the normal constraints are used. If timeLike = 0, the parameter is constrained by the time series.

Models All models currently available in ModelManager are listed in this table, which also contains their filenames.

- ModelID — This one-based index is the model identifier. For example, if there are three models in a study type then they will be indexed 1, 2 and 3.

• ModelName	This specifies the full model name. This name is the same for both linear and weighted models, e.g. “Zero Order”
• ModelAbbrevName	This specifies an abbreviated name for the model, which may be useful in report generation, e.g. “Parent Only” is abbreviated to “PO”.
• ModelUsed	Obsolete field. Must be 0.
• StudyTypeID	This is the identifier of the study type to which the model belongs.
• ModelFile	This field contains the filename of the model file.
• LogModelFile	This field contains the filename of the model file to be used for the weighted model.
• ModelBitmap	This field contains the filename of the model bitmap. This can be used to present the model equations in the reports.
• LogModelBitmap	Obsolete field. Must be empty.
• SearchID	This is the identifier of the search type to use for initial parameter estimation, 0 - Grid search, 1 - Simulated annealing.
• SearchExclusive	This specifies whether the search method is always used for the initial parameter estimation. 0 - only when specified by the user, 1 - always.
• ModelParamCalc	This is the name of the model component used to calculate the model’s initial parameter values.
• ModelDTXCalc	This is the name of the model component used to calculate the DT values for the model.
• ModelDTXType	Specifies the type of the component used to calculate the DTX values. 0 - Independent

event, 1 - Component event.

- ExtraModelData — This field contains any extra information that needs to be stored with this particular model. For models to be used in the parent with 2 metabolites study, this identifies the number of metabolites in the model (This will be 2 except under exceptional circumstances).

OptimMethods This table contains a list of all currently available methods of optimisation.

- OptimMethodName — This field contains the name of the optimisation method.
- OptimMethodID — This field contains the ID number of the optimisation method. The possible values are 0 (Marquardt) or 1 (Simplex).

Reports This table contains details of all currently available reports.

- ReportID — This field contains the ID number of the report.
- StudyTypeID — This field specifies the study type to which the report belongs.
- reportName — This field contains the name of the report (not the file name) that is displayed by ModelManager in the list of reports on the analysis configuration screen.
- reportFile — This field contains the file name of the report template that is opened by ModelManager when the report is requested.

Searches This table contains a list of all currently available parameter search methods.

- searchName — This field contains the name of the search method.
- searchID — This field contains the ID number of the search method.

TDistribution This table contains a table of values of the student t distribution for various degrees of freedom.

- DegreesOfFreedom — This field contains the number of degrees of freedom.
- TValue — This field contains the student t value for a specific number of degrees of freedom.

TimeUnits This table contains a list of all currently available time units.

- TimeUnit — This field contains the name of the time units.
- TimeUnitsID — This field contains the ID number of the time units.

10. ModelManager Script functions

Introduction

This chapter desribes the scripting langauge functions provided by Modelmanager to control the operation of the application, database and ModelMaker clculation engine.

Database Functions

DBOpen

```
Function DBOpen (
      ByVal alias As String,
      ByVal filename As String,
      ByVal SQL As String
) As Boolean
```

This function opens a database connection to a database file and executes an SQL statement to generate a record set. The record set is given an alias name which is used to access it in future.

Parameters

alias	This is a symbolic name for the record set generated by the SQL statement.
filename	This is the fully justified file name of the database file to be opened.
SQL	This is the SQL statement to be executed once the database connection has been established.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE.

DBClose

```
Function DBClose (
      ByVal alias As String
) As Boolean
```

This function closes a database connection.

Parameters

alias This is a symbolic name for the record set to be closed.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE.

DBGetFieldValue

```
Function DBGetFieldValue (
      ByVal alias As String,
      ByVal field As String
) As String
```

This function returns the contents of a named field from a record set.

Parameters

alias This is a symbolic name for the record set to be read from.

field This is the name of the field in the record set to be read.

Return Value

If the function succeeds, it returns the contents of the field as a string.

If the function fails, it returns an empty string.

DBGetColumnValue

```
Function DBGetColumnValue (
      ByVal alias As String,
      ByVal index As Integer
) As String
```

This function returns the contents of a field from a record set.

Parameters

alias	This is a symbolic name for the record set to be read from.
index	This is the zero-based index of the field in the record set to be read.

Return Value

If the function succeeds, it returns the contents of the field as a string.

If the function fails, it returns an empty string.

DBSetFieldValue

```
Function DBSetFieldValue (
      ByVal alias As String,
      ByVal field As String,
      ByVal value As String
) As Boolean
```

This function sets the contents of a named field from a record set.

Parameters

alias	This is a symbolic name for the record set to be written to.
field	This is the name of the field in the record set to be written to.

value This is the string representation of the value which is to be written to the field.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBSetColumnValue

```
Function DBSetColumnValue (
      ByVal alias As String,
      ByVal index As Integer,
      ByVal value As String
) As Boolean
```

This function sets the contents of a field from a record set.

Parameters

alias This is a symbolic name for the record set to be written to.

index This is the zero-based index of the field in the record set to be written to.

value This is the string representation of the value which is to be written to the field.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBRequery

```
Function DBRequery (
      ByVal alias As String
) As Boolean
```

This function refreshes the contents of a record set.

Parameters

alias This is a symbolic name for the record set to be refreshed.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBNext

```
Function DBNext (
      ByVal alias As String
) As Boolean
```

This function moves to the next record in a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails or an attempt to move beyond the end of a record set is made, it returns FALSE.

DBPrevious

```
Function DBPrevious (
      ByVal alias As String
) As Boolean
```

This function moves to the previous record in a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails or an attempt to move beyond the start of a record set is made, it returns FALSE.

DBEOF

```
Function DBEOF (
      ByVal alias As String
) As Boolean
```

This function tests whether a record set is at the last record.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the record set is at the last record, it returns TRUE.

If the function fails or the record set is not at the last record, it returns FALSE.

DBBOF

```
Function DBBOF (
      ByVal alias As String
) As Boolean
```

This function tests whether a record set is at the first record.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the record set is at the first record, it returns TRUE.

If the function fails or the record set is not at the first record, it returns FALSE.

DBTop

```
Function DBTop (
      ByVal alias As String
) As Boolean
```

This function moves to the top of a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails or the record set is empty, it returns FALSE.

DBBottom

```
Function DBBottom (
      ByVal alias As String
) As Boolean
```

This function moves to the bottom of a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBUpdate

```
Function DBUpdate (
      ByVal alias As String
) As Boolean
```

This function updates the database with the most recently written values. This function should not be called by the ModelManager scripts.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBNumColumns

```
Function DBNumColumns (
      ByVal alias As String
) As Integer
```

This function returns the number of columns (fields) in a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns the number of columns in the record set.

If the function fails, it returns the value -1.

DBNumRows

```
Function DBNumRows (
      ByVal alias As String
) As Integer
```

This function returns the number of rows (records) in a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns the number of rows in the record set.

If the function fails, it returns the value -1.

DBDeleteRow

```
Function DBDeleteRow (
      ByVal alias As String
) As Boolean
```

This function deletes the current row from a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBAddRow

```
Function DBAddRow (
      ByVal alias As String
) As Boolean
```

This function adds in empty row to the current row from a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBDeleteTable

```
Function DBDeleteTable (
      ByVal alias As String
) As Boolean
```

This function deletes the entire contents of a record set.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, it returns FALSE.

DBCloseAll

```
Sub DBCloseAll (

)
```

This function closes all the database connections.

DBFlush

```
Sub DBFlush (

)
```

This function flushes all the cached updates of a record set to the database.

DBEmptyTable

```
Function DBEmptyTable(

      ByVal alias As String

) As Boolean
```

This function tests whether a record set has any records present.

Parameters

alias This is a symbolic name for the record set to be operated on.

Return Value

If the record set is empty, it returns TRUE.

If the function fails or the record set contains records, it returns FALSE.

Calculation Engine

CEStartEngine

```
Function CEStartEngine (
      ByVal status As boolean)
As Boolean
```

This function sets up the Calculation Engine and must be called before any other procedures are called.

Parameters

status If TRUE, to calculation engine status dialog is shown.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEStopEngine

```
Sub CEStopEngine (
)
```

This function closes down the Calculation Engine and clears up any resources.

CELoadModel

```
Function CELoadModel (
      ByVal model As String,
      ByVal title as String)
As Boolean
```

This function loads the model with the given file name and prepares it for running.

Parameters

model	This is the file name of the model to load. If the file name does not contain a path then the Calculation Engine looks in the current directory.
title	This is the title of the model which is shown in the calculation engine dialog.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

Remark

If a model is already loaded, the function fails.

CEUnloadModel

```
Sub CEUnloadModel (

)
```

This function unloads the currently loaded model.

CERunModel

```
Function CERunModel (
      ByVal alias As String,
      ByVal filename As String,
      ByVal SQL As String,
      ByVal case As String)
As Boolean
```

This function runs the currently loaded model.

Parameters

alias	This is the database alias name to use for the database connection which will receive the model values
filename	This is the fully justified filename of the database file to open.
SQL	This is the SQL statement to be executed to generate the record set which is exported from the model run.
case	This is the case number used to generate the record set which is exported from the model run.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEBeginModel

```
Function CEBeginModel (
) As Boolean
```

This function sets up the calculation engine for the evaluation of individual components of the currently loaded model.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEEndModel

```
Function CEEndModel (
) As Boolean
```

This function clears up the calculation engine after the evaluation of individual components of the currently loaded model.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEEvaluateModel

```
Function CEEvaluateModel (

) As Integer
```

This function evaluates the currently loaded model without exporting the calculated values to the database. Once the model run is complete the functions CEGetComponent can be called to retrieve the value of selected components.

Return Value

If the model run completes without error, it returns 0.

If the model run is terminated by a ModelStop event action, it returns -1.

If the model run is terminated by a mathematical error, it returns a positive value. To get extended error information, call GetLastError or GetLastErrorText.

CEActivateEvent

```
Function CEActivateEvent (

      ByVal symbol As String,

      ByVal state as Boolean)

As Boolean
```

This function sets the activation state of an event in the currently loaded model.

Parameters

symbol	This is the name of the event whose activation state is to be changed.
state	If TRUE, the event is activated. If FALSE, the event is de-activated.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEHasComponent

```
Function CEHasComponent (
      ByVal symbol As String)
As Boolean
```

This function tests whether the currently loaded model has a particular component.

Parameters

symbol This is the name of the component to test for.

Return Value

If the function finds the component, it returns TRUE.

If the function fails, the return value is FALSE.

CEGetComponent

```
Function CEGetComponent (
      ByVal symbol As String)
As Double
```

This function retrieves the current value of a component in the currently loaded model.

Parameters

symbol This is the name of the component whose value is to be retrieved.

Return Value

If the function succeeds, it returns the value of the component.

If the function fails, the return value is 0. To get extended error information, call GetLastError or GetLastErrorText.

Remarks

It is only valid to retrieve the values of Compartment, Variable, Flow and Defined Value components. Attempting to retrieve the value of another component returns an error.

CESetComponent

```
Function CESetComponent (
      ByVal symbol As String,
      ByVal value As Double)
As Boolean
```

This function sets the current value of a component in the currently loaded model.

Parameters

symbol This is the name of the component to set.

value This is the value to which the component is to be set.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

Remarks

It is only valid to set the values of Compartment and Defined Value components. Attempting to set the value of another component returns an error.

CEEvaluateComponent

```
Function CEEvaluateComponent (
      ByVal symbol As String)
As Boolean
```

This function evaluates the equation of a component in the currently loaded model.

Parameters

symbol This is the name of the component to set.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

Remarks

It is only valid to set the values of Compartment and Defined Value components. Attempting to set the value of another component returns an error. Once the equation has been evaluated the function CEGetComponent can be called to retrieve the component value.

CEEvaluateEvent

```
Function CEEvaluateEvent (
      ByVal symbol As String)
As Boolean
```

This function evaluates the actions of an event in the currently loaded model.

Parameters

symbol This is the name of the event to evaluate.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEGetFit

```
Function CEGetFit (
      ByVal alias As String,
      ByVal filename As String,
      ByVal SQL As String,
      ByVal case As String,
      ByVal transform As Boolean)
As Boolean
```

CEOptimiseModel

```
Function CEOptimiseModel (
) As Boolean
```

This function optimizes the currently loaded model.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEGetOptimiseStat

```
Function CEGetOptimiseStat (
      ByVal index As Integer)
As Double
```

This function retrieves the optimization statistics from the most recent optimization of the currently loaded model.

Parameters

index This is the index of the optimization option to set. Valid indices are as follows:

0	Corrected model degrees of freedom.
1	Corrected model weighted sum of squares
2	Model mean square
3	Residual degrees of freedom
4	Residual weighted sum of squares
5	Residual mean square
6	Corrected total degrees of freedom
7	Corrected total weighted sum of squares
8	R^2
9	F-Value
10	P-Value
11	Q-Value
12	Uncorrected total weighted sum of squares
13	Uncorrected model weighted sum of squares
14	Uncorrected total degrees of freedom
15	Uncorrected model degrees of freedom

Return Value

If the function succeeds, it returns optimization statistic.

If the function fails, the return value is 0. To get extended error information, call GetLastError or GetLastErrorText.

CESetParameter

```
Function CESetParameter (
      ByVal name As String,
      ByVal index As Integer,
```

```
      ByVal value As Double)
As Boolean
```

This function sets a value of the parameter of the currently loaded model.

Parameters

name This is the name of the parameter to set.

Index This is the index of the value it set. The valid indices are as follows

0 The value of the parameter.

1 The error of the parameter.

2: The type of the constraint to use for the parameter. Valid values are:

0	Default
1	Range
2	Fixed

3: The constraint range value.

4: The type of the constraint range. Valid values are:

0	Mulitplicative
1	Absolute

5: The lower fixed constraint value.

6: The upper fixed constraint value.

7: The lower grid search value.

8: The upper grid search value.

9: The number of steps between the upper and lower grid search values

10: Whether to use logarithmic steps. Valid values are:

0	linear steps
1	logarithmic steps

11: Whether to use ascending steps. Valid values are:

0	ascending steps
1	descending steps

12: This is the type of the parameter distribution. Valid values are:

0	None
1	Uniform
2	Triangular
3	Normal
4	Log-normal
5	Exponential
6	Weibull

13: The distribution lower value.

14: The distribution upper value.

15: The distribution mean value.

16: The distribution standard distribution value.

17: The distribution mode value.

18: The optimized value of the parameter

19: The optimized error of the parameter

20: The minimized error of the parameter

21: The type of the grid to use for the parameter grid search. Valid values are:

0	Default
1	None
2	Range
3	Fixed

value This is the new value to which the appropriate parameter value is set.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEGetParameter

```
Function CEGetParameter (
      ByVal name As String,
      ByVal index As Integer)
As Double
```

This function retrieves the value of a parameter of the currently loaded model.

Parameters

name	This is a NULL terminated string containing the name of the parameter whose value is to be retrieved.
index	This is the index of the value it set. See CEGetParameter function for valid index values.
value	This is a pointer to the buffer to receive the requested value of the parameter.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CESetLookupData

```
Function CESetLookupData (
      ByVal symbol As String,
      ByVal page As Integer,
      ByVal alias As String,
```

```
      ByVal filename As String,

      ByVal SQL As String,

      ByVal transform As Boolean)

As Boolean
```

This function sets the values of a Lookup Table in the currently loaded model.

Parameters

symbol	This is the name of the Lookup Table whose series are to be updated.
alias	This is the database alias name to use for the database connection.
filename	This is the fully justified filename of the database file to open.
SQL	This is the SQL statement to be executed to generate the record set which is imported into the lookup table..
transform	If this is TRUE, the data read in is log transformed. If this is FALSE the raw data is imported.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CESetModelData

```
Function CESetModelData (

      ByVal page As Integer,

      ByVal alias As String,

      ByVal filename As String,

      ByVal SQL As String,
```

```
      ByVal case As String,

      ByVal  transform As Boolean)

As Boolean
```

This function sets the values of a Lookup Table series.

Parameters

page	This is the number of the page whose series is to be updated.
alias	This is the database alias name to use for the database connection.
filename	This is the fully justified filename of the database file to open.
SQL	This is the SQL statement to be executed to generate the record set which is imported into the lookup table..
case	This is the case number used to generate the record set which is imported into the lookup table..
transform	If this is TRUE, the data read in is log transformed. If this is FALSE the raw data is imported.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CESetRunOption

```
Function CESetRunOption (

      ByVal index As Integer,

      ByVal value As Double)

As Boolean
```

This function sets the specified run option for the currently loaded model.

Parameters

index This is the index of the run option to set. Valid indices are as follows:

0: The start value of the run

1: The stop value of the run

2: The number of output steps

3: The random seed for the run

4: The integration method to use. Value values are as follows:

0	Euler
1	Mid-point
2	Runge-Kutta
3	Bulirsch-Stoer

5: Whether the integration method if fixed step. Valid values are as follows:

0	Adaptive
1	Fixed

6: The accuracy value for the integration method.

7: The minimum value for the integration method.

8: The starting steps value for the integration method.

9: The fixed steps value for the integration method.

10: The error scaling type for the integration method. Valid values are as follows:

0	by the function
1	by the function and its derivative
2	by a constant value.

11: The error scaling value for the integration method.

12: The accuracy value for the run-time integration function.

13: The steps factor for the run-time integration function.

value This is the value to which the specified run option is set.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEGetRunOption

```
Function CEGetRunOption (
      ByVal index As Integer)
As Double
```

This function retrieves the specified run option for the currently loaded model.

Parameters

index This is the index of the run option to retrieve. See the CESetRunOption function for a list of valid indices.

Return Value

If the function succeeds, it returns the value of the run option.

If the function fails, the returned value is 0. To get extended error information, call GetLastError or GetLastErrorText.

CESetOptimiseOption

```
Function CESetOptimiseOption (
      ByVal index As Integer,
      ByVal value As Double)
As Boolean
```

This function sets the specified optimization option for the currently loaded model.

Parameters

index This is the index of the optimization option to set. Valid indices are as follows:

indices are as follows:

0: The method for the optimization. Valid values are as follows:

0	Marquardt
1	Simplex

1: Whether the method uses least squares optimization. Valid values are as follows:

0	Error weighted optimization
1	Least squares optimization

2: The number of convergence steps for the optimization.

3: The default constraint range value for the optimization.

4: The type of the default constraint range. Valid values are:

0	Mulitplicative
1	Absolute

5: The default data error value for the optimization.

6: The type of the default data error. Valid values are:

0	Fractional
1	Absolute

7: The initial lambda value for the optimization.

8: The convergence change value for the optimization.

9: The fractional change value for the optimization.

10: The minimum change value for the optimization.

11: The retry count value for the optimization.

12: The default number of grid steps to use for the optimization.

13: The initial parameter estimation type to use for the optimization. Valid values are:

0	None
1	Grid Search
2	Simulated Annealing

14: Whether to use the constraint equation during the optimization. Valid values are:

0	No constraint equation
1	Use constraint equation.

value This is the value to which the specified optimization option is set.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CEGetOptimiseOption

```
Function CEGetOptimiseOption (
      ByVal index As Integer)
As Double
```

This function retrieve the specified optimization option for the currently loaded model.

Parameters

index This is the index of the optimization option to retrieve. See CESetOptimiseOption for a list of valid indices.

Return Value

If the function succeeds, it returns the value of the optimization option.

If the function fails, the return value is 0 To get extended error information, call GetLastError or GetLastErrorText.

CESetOptimiseParameter

```
Function CESetOptimiseParameter (
      ByVal symbol As String,
     ByVal state As Boolean)
As Boolean
```

This function sets the optimization status of the given parameter for the currently loaded model.

Parameters

symbol This is the name of the parameter whose status is to be set.

state This is the state of the parameter. If it is set to TRUE the parameter will be included in the next optimization. If it is set to FALSE the parameter will not be included in the optimization process.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

CESetLookupPage

```
Function CESetLookupPage (
      ByVal symbol As String,
      ByVal index As Integer)
As Boolean
```

This function sets the active page of a Lookup Table in the currently loaded model.

Parameters

symbol This is the name of the Lookup Table whose page is to be set.

page This is the number of the page which is to be activated.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE. To get extended error information, call GetLastError or GetLastErrorText.

Remarks

It is only valid to set the page of Lookup Table.

CEGetLookupPage

```
Function CEGetLookupPage (
      ByVal symbol As String)
As Integer
```

This function retrieves the active page of a Lookup Table in the currently loaded model.

Parameters

symbol This is the name of the Lookup Table whose page is to be retrieved.

Return Value

If the function succeeds, it returns the active page number.

If the function fails, the return value is 0. To get extended error information, call GetLastError or GetLastErrorText.

Remarks

It is only valid to retrieve the page of Lookup Table.

CEGetLastError

```
Function CEGetLastError (
) As Integer
```

This function returns the number of the last error reported by any of the Calculation Engine functions.

Return Value

The error code of the last Calculation Engine function.

CEGetLastErrorText

```
Function CEGetLastErrorText (
) As String
```

This function returns the error text associated with the last error reported by any of the Calculation Engine functions.

Return Value

The last error text.

CEDebugMessage

```
Sub CEDebugMessage (
      ByVal message As String
)
```

This function displays a message in the debug window of the calculation engine status dialog.

Parameters

message This is the text to be displayed.

CEHasCovariance

```
Function CEHasCovariance (
      ByVal symbol1 As String,
      ByVal symbol2 As String
) As Boolean
```

This function tests whether there is an entry in the covariance table of the currently loaded model.

Parameters

symbol1 This is the symbol of the first component for which the entry is to be tested.

symbol2 This is the symbol of the second component for which the entry is to be tested.

Return Value

If a covariance entry exists, it returns TRUE.

If no covariance entry exists, it returns FALSE.

CEGetCovariance

```
Function CEGetCovariance (
      ByVal symbol1 As String,
      ByVal symbol2 As String
) As Double
```

This function gets a value from the covariance table of the currently loaded model.

Parameters

symbol1	This is the symbol of the first component for which the entry is to be retrieved.
symbol2	This is the symbol of the second component for which the entry is to be retrieved.

Return Value

If the function succeeds, it returns the covariance table value.

If the function fails, it returns the value -1.

CEGetParameterName

```
Function CEGetParameterName (
      ByVal index As Integer
) As String
```

This function gets the symbol of a model parameter of the currently loaded model.

Parameters

index	This is the zero-based index of the parameter symbol to retrieve.

Return Value

If the function succeeds, it returns the parameter symbol.

If the function fails, it returns an empty string.

CESetModelSeries

```
Function CESetModelSeries     (
      ByVal page As Integer,
      ByVal index As Integer,
      ByVal state As Boolean
) as Boolean
```

This function sets the used status of a model data series of the currently loaded model.

Parameters

- page — This is the number of the Model Data View page.
- index — This is the zero-based index of the series the status of which is to be changed.
- state — If set to TRUE, the series is set to be component data values. If set to FALSE, the series is set to unused.

Return Value

If the function succeeds, it returns TRUE.

If the function fails, the return value is FALSE.

CEGetComponentName

```
Function CEGetComponentName (
      ByVal index As Integer,
    ByVal type As Integer
) As String
```

This function gets the symbol of a component of the currently loaded model.

Parameters

index — This is the zero-based index of the component symbol to retrieve.

type — This is the type of the component to find. Valid values are:

1	Compartment
2	Variable
3	Defined Value

4	Flow
5	Influence
6	Delay
7	Parameter
8	Sub-model
9	Lookup File
10	Lookup Table
11	Independent Event
12	Component Event
13	DLL Function

Return Value

If the function succeeds, it returns the component symbol.

If the function fails, it returns an empty string.

CEGetDefinedName

```
Function CEGetDefinedName (
      ByVal symbol As Integer
) As String
```

This function gets the symbol of a defined value component of the currently loaded model.

Parameters

index This is the zero-based index of the defined value component symbol to retrieve.

Return Value

If the function succeeds, it returns the parameter symbol.

If the function fails, it returns an empty string.

Index

Activating ModelManager
  Licensing codes, 3
Adding Models, 77
Adding Reports, 75
Analysis Configuration, 33
Bitmaps, 58
Configure Parameters, 35
Contact Cherwell Scientific, 4
Data imports
  Offset, 50
Default Parameter Values, 88
DT Calculation, 100
Events, 104
Experimental Data Entry, 29
Final Report Format, 51
Formatting text, 53
Graph Screen, 31
Keyboard shortcuts, 7
Model Bitmap, 80
Model Bitmaps, 108
Model Export, 94
Model Import, 96
Model Parameters, 84
ModelMaker Models, 99
ModelManager Basics, 7
ModelManager Databases, 111
ModelManager screens
  details, 27
Models
  Parent and Metabolite, 103
  Parent Only, 102
  Parent with Multiple Applications, 102
  Sediment Partitioning, 104
ModelXchange, 42, 93
Optimization Configuration, 40
Output Report
  Configuration, 53
Parameter
  mO, 100
Parameter Aliases, 86
Parameter Estimation, 99
Parameter Search Configuration, 82

Passwords, 70, 107
  Database, 112
Presentation Sheets
  Format, 56
Report Generation Macros
  Pseudocode, 61
Report Worksheets, 45
Run Configuration, 38
Screens
  Analysis Configuration, 33
  Bitmap File, 81
  Configure Parameters, 35
  Default Parameter Values, 89
  Experimental Data Entry, 29
  Graph, 31
  Model Files, 78
  Model Parameters, 85
  Optimization Configuration, 40
  Parameter Alias, 87
  Parameter Search Configuration, 83
  Run Configuration, 38
  Study Details, 28
  Study Type, 27
  View Existing Study, 21
Script functions, 143
Settings Dialog, 42
Shortcut Bar, 11
Study Details, 28
Study Type, 27
The Filenames Sheet, 51
The General Sheet, 46
Tutorial, 8
  creating a new study, 11
Variable Names, 105
Visual Basic Macros, 59
Weighting Models, 107
Wizard, 73
Worksheets, 70