

mathwise

Courseware Management System
Version 2.1



Customising Mathwise Tutorial

Contents

Chapter 1	<i>Introduction</i>	3
	Notes on examples	3
Chapter 2	<i>Multiple Mathwise Setups</i>	4
	How to do it — Windows 3.1	4
	How to do it — Windows 95/NT	5
	How to do it — Other program launchers	5
Chapter 3	<i>Specifying Modules</i>	6
	How to do it	6
Chapter 4	<i>Adding your own Material</i>	9
	Auto-indexing	9
	How to do it	10
Chapter 5	<i>Adding New File Types</i>	13
	How to do it	13
Chapter 6	<i>Customising the Entry Unit</i>	16
	Overriding the default entry unit initialisation	16
	Specifying 'special' items	16
	How to do it	17
	Other entry unit customisation	18
Chapter 7	<i>Putting it all Together</i>	19
	The Scenario	19
	The Steps	19
	The Solution	20
Appendix 1	<i>Long File names</i>	23

Mathwise Courseware Management System v2.1 Customising Mathwise Tutorial
©Copyright 1998 by The University of Birmingham. All rights reserved.

Chapter 1

Introduction

One of the most powerful features of Mathwise is its ability to be tailored and modified to suit any given teaching needs. This document describes the main customisation features of Mathwise and gives you clear worked examples to show you how to take advantage of them.

Notes on examples

In all the examples in this document we will assume that the Mathwise system directory (i.e., where all the main Mathwise files such as `mathwise.exe` are stored) is `c:\mw`. Replace this with the location of your own Mathwise system directory if you are working through the examples.

Chapter 2

Multiple Mathwise Setups

One of Mathwise's most powerful customisation features is its ability to specify a variety of different system setups without copying any additional material from the CD. A typical use of this would be to present different choices of modules appropriate to different courses or years.

The key to using this feature is Mathwise's ability to override the default initialisation information that is created when Mathwise is first installed. This information is contained within the file `mw.ini`, which is a simple ASCII text file. When you run Mathwise this file is read by the system in order to determine which modules to load, which programs to use to display the modules. However, it is possible to create a new version of this file containing different setup information, let us call it `my_mw.ini` for example, and specify that Mathwise should use that file rather than `mw.ini` when it is run. We call this a **custom setup**. You can create as many custom setups as you need; since they only require files with different setup information, they will not in any way affect your basic Mathwise installation. For example, you may require a custom setup which specifies what material the user has access to during an assessment.

How to do it

The method for overriding the default initialisation depends on which version of the Windows operating system you are running.

Windows 3.1

For Windows 3.1 we will create a new Program Manager item that uses the new configuration file `my_mw.ini`.

- Create `my_mw.ini` by making a copy of `mw.ini` and renaming it to `my_mw.ini`. It should remain in the Mathwise system directory.
- If you do not already have one, create a new Program Manager group called **Mathwise** (select **File | New...** Choose **Program Group** and click **OK**. Type `MATHWISE` in the **Description** box and click **OK**).
- Open the Mathwise group and bring up the New Item dialog (select **File | New...** Choose **Program Item** and click **OK**).
- Type a name for the item in the **Description** box. Call it **Mathwise Custom**. You are free to specify any name, but we will use this name for now.
- Click in the **Command Line** box and then click the **Browse...** button. Use the **Browse** dialog box to locate `mathwise.exe` within your Mathwise system directory. Select it and click **OK**. The **Command Line** box will now read something like `c:\mw\mathwise.exe`. The override command is added to the end of this.
- To the end of the command line you should add `-I` (this means 'override the default initialisation') and the full path to `my_mw.ini`. In this example it will therefore look like this:

```
c:\mw\mathwise.exe -Ic:\mw\my_mw.ini
```

Note that there is a space before the `-I` and that there is no space between `-I` and the file-path. Also it is not necessary to give the full path to the new initialisation file if it is in the Mathwise system directory — we have done it here for the sake of clarity.

- You must also set the **Working Directory** to point to the Mathwise system directory. Type `c:\mw` in the edit box.
- Finally click on **OK** to close the **Program Item Properties** dialog and create the new item which will appear as a new icon, with label `MATHWISE (CUSTOM)`, in the **Mathwise** group.

- You can now run Mathwise with the `my_mw.ini` override by double-clicking on the **Mathwise Custom** icon. Of course since the custom setup is just a copy of the default you will not notice any difference. We will be modifying `my_mw.ini`, so that you will see a difference, in later chapters of this tutorial.

Windows 95/NT

For Windows 95/NT we will create a new shortcut to Mathwise that uses the new configuration file `my_mw.ini`.

- Create `my_mw.ini` by making a copy of `mw.ini` and renaming it to `my_mw.ini`. It should remain in the Mathwise system directory.
- Now, using Explorer, locate `mathwise.exe` within the Mathwise system directory and create a new shortcut to it (either right-click on `mathwise.exe` and select **Create Shortcut** from the pop-up menu, or select `mathwise.exe` and choose **File | Create Shortcut** from the Explorer menu).
- You will now have a shortcut file called **Shortcut to mathwise.exe**. Rename this to **Mathwise Custom**. You are free to specify any name, but we will use this name for now.
- We now need to alter the Properties of the **Mathwise Custom** shortcut. Bring up the Properties dialog for **Mathwise Custom** (either right-click on **Mathwise Custom** and select **Properties** from the pop-up menu; or select **Mathwise Custom** and choose **File | Properties** from the Explorer menu).
- Click on the **Shortcut** tab of the dialog and move to the **Target** edit box. You will see the command line that is run when the shortcut is double-clicked (something like `c:\mw\mathwise.exe`). It is here that we will add the override command.
- To the end of the command line you should add `-I` (this means ‘override the default initialisation’) and the full path to `my_mw.ini`. In this example it will therefore look like this:

```
c:\mw\mathwise.exe -Ic:\mw\my_mw.ini
```

Note that there is a space before the `-I` and that there is no space between `-I` and the file-path. Also it is not necessary to give the full path to the new initialisation file if it is in the Mathwise system directory — we have done it here for the sake of clarity.

- You can now run Mathwise with the `my_mw.ini` override by double-clicking the **Mathwise Custom** shortcut. This shortcut can be moved or copied to the Start Menu or onto the desktop if required. Of course since the custom setup is just a copy of the default you will not notice any difference. We will be modifying `my_mw.ini`, so that you will see a difference, in later chapters of this tutorial.

Other program launchers

It is possible that, instead of using standard Windows features such as Program Manager items or Shortcut items to launch Mathwise, you are using a non-standard launcher, such as a menuing system or a custom shell. This is particularly likely for network installations. This should present no problem to using the Mathwise initialisation override feature. All that should be remembered is that placing `-I` and the name of the override file after the command-line that runs `mathwise.exe` effects the override.

Chapter 3

Specifying Modules

Perhaps the most useful aspect of using custom setups (as described in **Chapter 2**) is the ability to specify a sub-set of Mathwise modules. If you have a Mathwise installation containing many modules (the full version of Mathwise will have 52 modules) it can be confusing for students to have to choose the right module from a long list. You can easily trim the list down to just those modules that a particular group of students needs to use, and create a Program Manager item or Shortcut specifically for those students. Note that this will not prevent access to the full range of modules from different custom setups.

How to do it

To pick a sub-set of modules for the custom setup we will edit the `my_mw.ini` override initialisation file we created in **Chapter 2**.

- Locate `my_mw.ini` in the Mathwise system directory and open it with a text editor such as Notepad.
- Scroll down the file until you reach the section headed `[modules]`. It is here that we specify which modules (and other material) to which users will have access. This is done by means of a list of Mathwise Index Files. These files, which have a suffix `MWX`, tell Mathwise about courseware material. Typically each module has three associated `MWX` files. One of these describes the main module, learning units and module resources and is usually named with the module code (for example `m1lineq.mwx` for the Inequalities module). Another of the `MWX` files describes the module's leaflets (and is usually named with the module-code and `_l` appended, e.g. `m1lineq_l.mwx`). The remaining file describes the module's assessment (and is usually named with the module code and `t` appended, e.g., `m1lineqt.mwx`.)
- The key therefore to specifying which modules a custom setup contains is simple: only those modules whose index files are included in this section will be available. Bearing this in mind let us specify a subset of modules for `my_mw.ini`
- Bearing this in mind let us specify a sub-set of modules for `my_mw.ini`.
- For brevity we will assume that your Mathwise installation only contains a few modules — `M1GRPH`, `M1INEQ`, `M1LOGS`, `M2TFNS`, `E1CPLX` and `E4VECT`. The `[modules]` section would therefore currently look like this:

```
[modules]
EntryUnit=ent_unit.mwx
mwtools=mwtools.mwx

m1grph=modules\m1grph\m1grph.mwx
m1grphLeafs=modules\m1grph\m1grph_l.mwx

m1lineq=modules\m1lineq\m1lineq.mwx
m1lineqLeafs=modules\m1lineq\m1lineq_l.mwx
m1lineqTest=modules\m1lineq\m1lineqt\m1lineqt.mwx

m1logs=modules\m1logs\m1logs.mwx
m1logsLeafs=modules\m1logs\m1logs_l.mwx
m1logsTest=modules\m1logs\m1logst\m1logst.mwx

m2tfns=modules\m2tfns\m2tfns.mwx
m2tfnsLeafs=modules\m2tfns\m2tfns_l.mwx
m2tfnsTest=modules\m2tfns\m2tfnst\m2tfnst.mwx
```



```
m2tfns=modules\m2tfns\m2tfns.mwx
m2tfnsLeafs=modules\m2tfns\m2tfns_1.mwx
m2tfnsTest=modules\m2tfns\m2tfnst\m2tfnst.mwx

elcplxLeafs=modules\elcplx\elcplx_1.mwx

e4vect=modules\e4vect\e4vect.mwx
e4vectLeafs=modules\e4vect\e4vect_1.mwx
e4vectTest=modules\e4vect\e4vectt\e4vectt.mwx

blbiog=modules\blbiog\blbiog.mwx
```

- Try this modification and save `my_mw.ini`. Run **Mathwise Custom** again and you will still see only the trigonometry and vectors modules listed but now any links within those modules to external leaflets will work.

In conclusion you can see that specifying which material you wish users to have access to is a simple matter: make a new initialisation file by copying `mw.ini`, remove all references (except for leaflets) to the modules you do not require from the `[modules]` section, and then create a new item or shortcut to run `mathwise.exe` using the new initialisation file.

Chapter 4

Adding your own Material

Another powerful feature of Mathwise is the ability to add your own material to the system so that it can be easily accessed by students throughout the Mathwise environment. Examples of material you may wish to add could include the following:

- course notes that tell students what courseware they should be concentrating on
- sample exam questions
- your own worksheets
- background documentation on different topics
- links to web pages
- other multimedia teaching software
- your own courseware material

The Mathwise system can accommodate all these and allow you to make the material accessible in a variety of ways.

Note that the current version of Mathwise (v2.1) cannot go directly to http links, it can only run files. If you want students to be able to access a selection of Web sites you should create your own HTML file locally (say `sites.htm`) which contains the http links and allow Mathwise to run that file using your chosen browser. The student can then follow the links from `sites.htm`.

There are basically two ways of adding your own material to the system: the **auto-indexing** method, which is straightforward but allows for less flexibility than the second method, **manual indexing**. For nearly all eventualities the auto-indexing method should be adequate for your needs and it is this method that we shall deal with in this tutorial. For information on manual indexing, which involves creating your own Mathwise index files, you should consult chapter 3 of the **Courseware Manager** manual, provided on the Mathwise web site at <http://www.bham.ac.uk/mathwise/>.

How students access the material depends to some extent on which of these two methods you use. It is possible with both methods to give access to the material (and your own selection of existing Mathwise courseware) from the 'Special' screen of the Mathwise entry unit. With auto-indexing you can also allow access to the material via the Mathwise resource browser from anywhere within the Mathwise courseware.

Auto-indexing

Auto-indexing involves little more than adding a reference to the files you wish to include in a Mathwise initialisation file (this can be the default `mw.ini` or a custom setup file) and specifying how you want students to be able to access them. In addition, if you wish to add material of a type that is not supported by the default installation of Mathwise, you will also need to tell the system how to run it.


***Technical Note:** What 'auto-indexing' means is essentially that Mathwise will create a virtual Mathwise Index File (MWX file — see **Chapter 3**) at run-time for the auto-indexed material.*

As an example we will add four items to the custom setup (`my_mw.ini`) we have been working on in earlier chapters.

1. Some course notes for Years 1 and 2 which are in simple ASCII text files (a file type which is supported by default). We shall assume these are located in e:\students\ref as cnotes1.txt and cnotes2.txt.
2. An Excel spreadsheet file containing a worksheet (a non-supported file type) found at e:\students\work\works1.xls.
3. A link to an HTML file (a non-supported file type) found at the University X website at n:\website\start.htm.

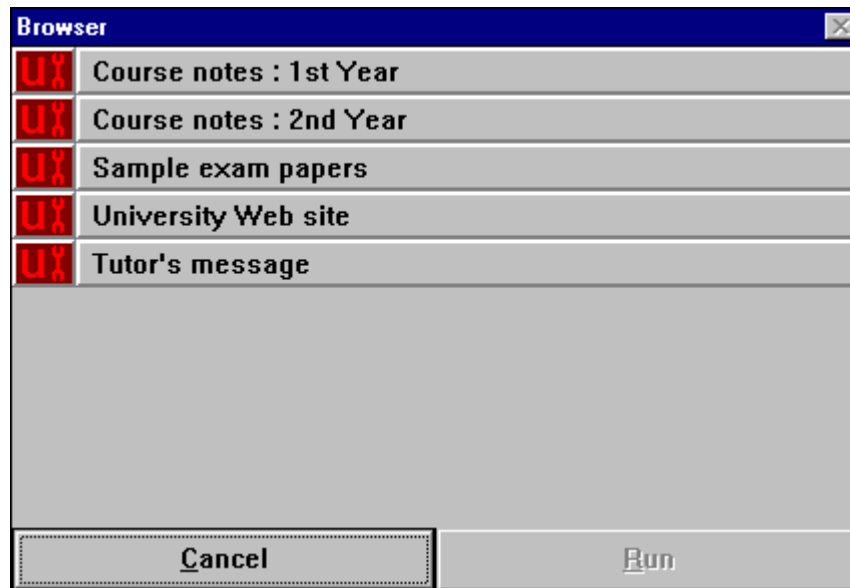
Note that if you are working through these examples you should, of course, substitute some material of your own and use the examples as a guideline.

How to do it

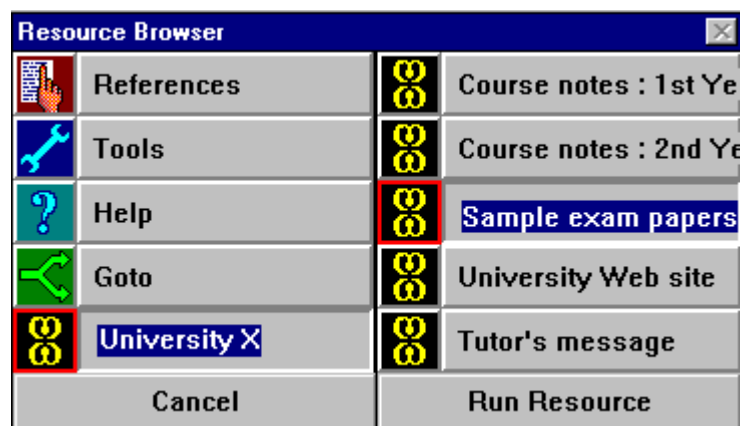
- Open the custom setup file my_mw.ini in a text editor.
- Locate the [autoindex] section of the file.
- You will see that this section contains a number of entries. Some of these (useAutoIndex, addToPalette, code and name) are used to tell the system how to make the auto-indexed material available. All other entries in this section are used to specify the actual material to auto-index.
- We will deal with the required entries first before adding the actual material.
- Firstly you should ensure that the useAutoIndex entry is set to 1, i.e., useAutoIndex=1. This simply switches on the auto-indexing feature. A value other than 1 disables the feature.
- Next you should set a value for name. The text you specify for this entry is the label that will appear on the Mathwise resource browser that gives access to the material from anywhere within the Mathwise courseware. You will most likely want to set this to the name of your institution. In this example we will set it to 'University X', i.e., name=University X. If you do not set a value for name then the word 'Special' will be used by default.
- Next we shall modify the addToPalette entry. This entry specifies how the auto-indexed material is available via the resource browser (this is the floating dialog box that appears by clicking the  button within Mathwise courseware). If it is set to 0 then the material will not be available via the resource browser. If you set it to 1 then a button will be added to the left-hand column of the browser, with a label of name and the Mathwise icon, e.g.



When clicked this will bring up a second floating list of the auto-indexed material from which the student can select an item to run, e.g.



If you set `addToPalette` to 2 then the name button will appear on the resource browser as before. This time, however clicking it will bring up a list of the auto-indexed material in the right-hand column of the resource browser from which the user can select an item to run e.g.



As a rough guideline on whether to choose 1 or 2 as the `addToPalette` value, 1 works best for a large amount of auto-indexed items, 2 for a small number. As we are only adding four items in this example we will therefore set it to 2, i.e., `addToPalette=2`

- The final 'how-to' entry in `[autoindex]` is `code`. As explained in the Technical Note earlier in this chapter, auto-indexing generates a virtual index-file for the auto-indexed files at runtime. By default it assigns a module code of `AUTO` to all these items. You can override this by setting `code` to some other value, but will rarely need to do this so and can generally just delete this `code` entry (if present).
- Finally we can add the new material. This is easy. For each file you wish to add simply type some short descriptive text for the file, then an 'equals' sign and the path to the file. Note that you cannot include '=' in the description! After including the values we set above for our example files this means the `[autoindex]` section of `my_mw.ini` will look something like this:

```
[autoindex]
useAutoIndex=1
name=University X
addToPalette=2
Course notes : 1st Year=e:\students\ref\cnotes1.txt
Course notes : 2nd Year=e:\students\ref\cnotes2.txt
Worksheet=e:\students\work\works1.xls
University Web site=n:\website\start.htm
```

Note that if you are adding your own equivalent material this will look a bit different.

- Now save the `my_mw.ini` and run **Mathwise Custom**. Run the Vectors module and then click on the resource browser button. You will see a button at the bottom left of the browser labeled `UNIVERSITY X`. Click it and a list of the added items will appear on the right. Click on one of the text file items (e.g., **Course notes: 1st Year** in this example) and the file should run (using Notepad). Close the file, bring up the browser again and this time select and try to run the **Worksheet** item. It does not run because we have not yet told Mathwise how to run files of this type. We will do this in **Chapter 5 — Adding New File Types**, which comes next.
- In addition you will observe that the new material you have just added is (probably) not yet available directly from the entry unit. Since material accessible directly from the entry unit can include other items which are not auto-indexed, we will deal with this in a separate chapter, along with other customisation of the entry unit. See **Chapter 6 — Customising the Entry Unit** for further details.

Special note: the filing systems of Windows 95 and NT both allow long file names (i.e., directory and file names can be longer than the 8 character limit imposed in MS-DOS and Windows 3.1). Because Mathwise is a 16-bit system, so that Windows 3.1 users can run it, it does not support long file names. If you try to auto-index a file whose path contains long file names then it will not run. A work around for this problem is described in **Appendix 1**.

Chapter 5

Adding New File Types

The standard installation of Mathwise is configured to support a range of file types. These are as follows:

- Executable files (*.exe)
- Asymetrix Toolbook™ files of versions 1.53, 3 and 4 (*.tbk, *.tb3, *.tb4, *.sbk)
- Macromedia Authorware™ files of versions 2 and 4 (*.app, *.apw, *.a4r)
- Windows Help files (*.hlp)
- ASCII text files (*.txt)

Mathwise can run material of these types because the system initialisation file `mw.ini` (or a custom setup override of this) contains information relating each file type (identified by its file suffix) to a particular **reader application**, that is to say an application capable of running that type of file. This relationship is known as a **file association**. So for example in the standard Mathwise installation Toolbook 1.53 files (i.e., those with suffix `tbk`) are associated with the `tbook.exe` reader application, supplied with Mathwise. When a user clicks a link in an item of Mathwise courseware the system checks the suffix of the file being requested, then looks in its internal list of file associations for a reader application corresponding to this suffix and, if this exists, runs the file with the defined application. If the file `ml1neq1.tbk` were requested from a link, it would be run using `tbook.exe`.

It is possible to both add and modify the list of file associations. There are two main reasons for doing this.

1. You have added new material to Mathwise (as we did in **Chapter 4**) of a type that is not supported by the standard Mathwise installation. For example you may include Microsoft Excel spreadsheet files which will require you to specify where Mathwise can find the Excel application.
2. You wish to change the default reader application for a particular file type. For example you may wish to run ASCII text files using your own editor program rather than the Notepad, which is the default reader application for *.txt files.

How to do it

As an example we will add two file associations to deal with the new file types we added in **Chapter 4**. You may recall that we used the auto-index feature to include an Excel spreadsheet file and an HTML document. Since neither of these is a standard Mathwise file-type, a reader application will need to be defined for each of them.

- Open the custom setup file `my_mw.ini` that we have been working with in the previous chapters in a text editor such as Notepad.
- Locate the `[associations]` section of the file.
- It should look something like this (omitting some entries for brevity):

```
[associations]
Assoc1Name=Toolbook
Assoc1Ext=tbk
Assoc1Path=tbook.exe
Assoc2Name=Authorware
Assoc2Ext=app
Assoc2Path=runawp2.exe
...
```

```
Assoc9Ext=tb4
Assoc9Path=tb40run.exe
Assoc10Name=Authorware
Assoc10Ext=apr
Assoc10Path=runa4w32.exe
```

There is also an optional entry `AssocTrack=0` or `1` (`1` by default)

- You will see that an association must be defined by the three entries: `AssocXName`, `AssocXExt` and `AssocXPath` (where `X` is an integer). To add a new association you need to add entries of each of these types, replacing `X` with the next number after the highest in the current association list – for example, the new number here will be `11`. `AssocTrack` takes an argument of `0` or `1`. Unless this is to be set to `0` there is no need to include it. When set to `0`, this option prevents the Mathwise system from tracking items once they are launched; making it effectively a ‘launch and forget’ option. We recommend this option if you intend to run Microsoft Internet Explorer v3 & later and Windows Help on NT, in both cases Mathwise may hang when the application is launched. As an example, the Association required to run Internet Explorer would read

```
Assoc12Name=HTML
Assoc12Ext=htm
Assoc12Path=c:\progra~1\intern~1\iexplore.exe
Assoc12Track=0
```

Let us now add an association for Excel files.

- Firstly add an entry `Assoc11Name` at the end of the `[associations]` section, and set it to `Excel`, i.e., `Assoc11Name=Excel`. This entry simply gives Mathwise a description of the type of the file being associated.
- Next add an entry `Assoc11Ext` and set it to `xls`, i.e., `Assoc11Ext=xls`. This entry specifies the file extension that identifies Excel files – for example, the file `works1.xls` which we added in **Chapter 4**.
- Finally add an entry `Assoc11Path` and set it to the location of the Excel application. This is where we tell Mathwise what reader application to use for this particular file type. If we assume that the Excel application resides at `c:\apps\excel\excel.exe` then the entry will look like this:
`Assoc11Path=c:\apps\excel\excel.exe`.
- We also need to add an association for HTML files, which usually have the extension `htm`. Let us suppose that Netscape Navigator is the application used to read HTML files, and it can be found at `c:\apps\browsers\netscape\netscape.exe`. The three entries should be added in the same fashion as the Excel entries but using the number `12`.
- The `[associations]` section of `my_mw.ini` should now look like this:

```
[associations]
Assoc1Name=Toolbook
Assoc1Ext=tbk
Assoc1Path=tbook.exe
Assoc2Name=Authorware
Assoc2Ext=app
Assoc2Path=runawp2.exe
...
```

```
Assoc9Ext=tb4
Assoc9Path=tb40run.exe
Assoc10Name=Authorware
Assoc10Ext=apr
Assoc10Path=runa4w32.exe
Assoc11Name=Excel
Assoc11Ext=xls
Assoc11Path=c:\apps\excel\excel.exe
Assoc12Name=HTML
Assoc12Ext=htm
Assoc12Path=c:\apps\browsers\netscape\netscape.exe
```

- Now save and close my_mw.ini and run **MATHWISE CUSTOM** again. Run a module, bring up the resource browser and try to run the Excel file and the HTML file we added in **Chapter 4**. All being well they should now run.

Special note: The filing systems of Windows 95 and NT both allow long file names (i.e., directory and file names can be longer than the 8 character limit imposed in MS-DOS and Windows 3.1). Because Mathwise is a 16-bit system, so that Windows 3.1 users can run it, it does not support long file names. If you try to add an association where the reader application path contains long file names it will not run. A work around for this problem is described in **Appendix 1**.

Chapter 6

Customising the Entry Unit

The entry unit is the front end of the Mathwise system, where users can select which module they wish to study, browse through learning-units and leaflets, and search the courseware. It can be customised in a similar way to the overall Mathwise system overall in previous chapters.

The entry unit, like the Mathwise system, has an initialisation file: by default this is `entryu.ini`. We will first see how to override the default. You may wish to do this if for example you have designed a series of custom setups for different courses and you want to tailor the entry unit for each course. (Changing the default `entryu.ini` would have a global effect across all the custom setups.)

Overriding the default entry unit initialisation

To override the default `entryu.ini` is relatively straightforward. We shall do it for the `MATHWISE_CUSTOM` setup that we have been working on throughout this tutorial.

- Make a copy of `entryu.ini` and call it `my_eu.ini`. Save it in the Mathwise system directory.
- Now open `my_mw.ini` and locate the `[general]` section at the top of this file.
- Search the section and find the `StartupDesc` entry. It should look like this: `StartupDesc=EUNT,,,`. This is a **Mathwise descriptor query** and basically means ‘start Mathwise with the standard entry unit’. (Further documentation of this level of the system is available in the **Courseware Manager** manual, which can be found on the Mathwise website at <http://www.bham.ac.uk/mathwise>.)
- As things stand when the entry unit starts it will use `entryu.ini` as the source of its initialisation data. We can override this by changing the `StartupDesc` entry to:

```
StartupDesc=EUNT,,,,_C:\MW\MY_EU.INI
```

- This entry now means ‘start Mathwise with the standard entry unit and pass it the `my_eu.ini` file to use as its initialisation data’.
- Save `my_mw.ini` and run **Mathwise Custom**. The entry unit looks exactly the same as before! This is of course because the override `my_eu.ini` is just a copy of the default that the entry unit has been using up until now. In the later sections in this chapter we shall modify `my_eu.ini` so that you will see a difference.

Specifying ‘special’ items

The ‘special’ screen of the entry unit is where courseware items specified in the entry unit initialisation file are listed. It is accessed via the penultimate button on the main menu, (whose label may also be specified, but which is **Special** by default):



How to do it

- Open the `my_eu.ini` file we created earlier in this chapter in a text editor.
- Locate the `[special]` section within the file.
- There are two entries in the section which tell Mathwise how to use the ‘special’ feature of the entry unit. The first of these is `useSpecial`. You should set this to 1, which simply turns the feature on; i.e., `useSpecial=1`. Any other value will turn the feature off and the ‘special’ button will be disabled.
- The other entry is `buttonName` and is the caption which replaces the default value of `Special`, on the button, and will appear as the title of the ‘special’ screen. In general you should set this to the name of your institution. Since we are using the ‘special’ screen to display the items we added in **Chapter 4** we will set this entry to the name of our hypothetical university i.e., `buttonName=University X`.
- Now we shall add the criterion necessary to display the items we added to Mathwise using the auto-index feature. After the name entry you should type: `desc1=UTIL,AUTO`
- To fully understand what this entry means you will have to understand the **Mathwise descriptor system** (this is the basis of how Mathwise courseware is indexed) which is described fully in the **Courseware Manager** manual provided on the Mathwise website at <http://www.bham.ac.uk/mathwise>. Put simply, the entry specifies a search criterion, known as a **descriptor query**, which means ‘list all items within Mathwise of type UTIL that have a module code of AUTO’. This is exactly how the files that are listed in `[autoindex]` are categorised — they are automatically indexed as utilities (i.e., of type UTIL) and are assigned a module code of AUTO.
- Note that it is possible to add any number of descriptor queries in the `[special]` section so that you can list any selection of courseware material you require on the ‘special’ screen. You would do this by adding new entries of the form `descX=dquery` (where X is a number, rising consecutively) and `dquery` is a descriptor query specifying the courseware you want listed. For example you might also wish, for some reason, to include a list of leaflets that have been key-worded with ‘`trig*`’ in which case you would add a second `descX` entry thus: `desc2=LEAF,,,,trig`. Do not do this for the example. To use this feature to its fullest extent is beyond the scope of this tutorial and you will have to study the **Courseware Manager** manual.
- When you are finished the `[special]` section should look like this:

```
[special]
useSpecial=1
buttonName=University X
desc1=UTIL,AUTO
```

- Save the `my_eu.ini` file and run **Mathwise Custom**. Click the **University X** button and you will see a screen listing the items you added in **Chapter 4**. Click on one to test that it actually

works.

Other entry unit customisation

You may wish to experiment with other possible customisations of the entry unit using **my_eu.ini**. These are as follows.

1. **Enable/disable the prompt field.** Moving the cursor over buttons and list boxes in the entry unit causes hint text to appear beneath the Mathwise logo, describing the functionality of these items. This can be turned off in the [general] section in an entry unit initialisation file by setting the showPrompt entry to a value other than 1. A value of 1 will enable the prompt.
2. **Showing Mathwise module codes.** In the modules screen of the entry unit (click **Courseware | Modules**) a six-character code appears before the name of each module. These codes are included by default because some modules refer to others by this code rather than by name. You can however control the display of these with various entries in the [general] section.
 - Setting showCodes to 1 will show the codes, any other value will hide them.
 - Setting codesColumn to 0 will display the code then the module name. A value of 1 will reverse the order.
 - The value of columnSpace determines how many pixels from the left-hand edge of the list box the second column is placed.
 - The contents of the [modulecodes] section of an entry unit initialisation file relates the internal module code with the text displayed in the code column. Do not edit this section.
3. **Showing descriptor queries.** If the debugShowDesc value in [general] is set to 1, trying to run any item from the entry will display the Mathwise descriptor query being used to select the item rather than running it. This can be useful for advanced configuration of the system. For example you may wish to include specific leaflets in the 'special' section; by using debugShowDesc you can locate the leaflets via **Courseware | Leaflets**, note down the descriptor queries required to run them, then add them to [special] using the descX entries (see above).
4. **Other entries.** These should not be altered.

Chapter 7

Putting it all Together

In this final chapter of the tutorial we shall use what we have learnt in the previous chapters to implement a real-life customisation of Mathwise. We will not go into great detail here, since all that is covered earlier in the tutorial, but will outline the steps needed, and at the end of the chapter present the relevant parts of the various initialisation files produced.

The Scenario

Mathwise has been installed on your Windows 3.1 network at University X with the Mathwise system located in `c:\mw` and the standard Mathwise modules in `e:\modules` (as seen from the client machines). The Mathwise modules installed are as follows: M1GRPH, M1INEQ, M1BASE, M1LOGS, M2TFNS, E1CPLX and E4VECT. These are all available to Mathematics students under the **Mathwise** icon created in the **Mathwise** program group.

You want to create custom setups for some support teaching you have to give using Mathwise material. One course is for first year sociology students who need to study basic algebra (M1BASE) and graphs (M1GRPH) as a basis for statistics teaching in their second year. The other course is a refresher for physics students due to study quantum mechanics. They need to brush up on trigonometry (M2TFNS), complex numbers (E1CPLX) and vectors (E4VECT).

Each course also requires some support material to be made available.

- Mathematics students are offered old exam papers, which are made available as Word documents: `c:\support\maths\mexam11.doc` and `c:\support\maths\mexam12.doc`.
- The sociology students also require Word exam papers (`c:\support\soc\s_exam1.doc` and `c:\support\soc\s_exam2.doc`) plus a link to the sociology department's webpages at `s:\www\soc\start.htm`.
- The physics students are offered some notes in a text file (`c:\support\physics\cnotes1.txt`) and also a list of **all** available Mathwise leaflets that address complex numbers (not just those in E1CPLX) and trigonometry (not just those in M2TFNS).

The Word application can be found at `f:\apps\mw\word\winword.exe` and the HTML browser at `f:\apps\browsers\netscape\netscape.exe`.

The Steps

The steps you should carry out to customise Mathwise to deal with this scenario are as follows.

- Make two copies of the `c:\mw\mw.ini` file and call them `c:\mw\mw_soc.ini` and `c:\mw\mw_phys.ini` respectively.
- Create two new program manager items within the **Mathwise** group and call them **Sociology** and **Physics** respectively (see **Chapter 2**). The **Command Line** for these items should read:
`c:\mw\mathwise.exe -Ic:\mw\mw_soc.ini`
and
`c:\mw\mathwise.exe -Ic:\mw\mw_phys.ini`
respectively. In both cases the **Working Directory** is `c:\mw`.
- Now you need to define the modules to be made available (see **Chapter 3**). In `mw_soc.ini` remove all references in the `[modules]` section to MWX files for modules and assessments other than M1BASE and M1GRPH; retain the system MWX files (`mwtools.mwx`, `ent_unit.mwx` and `blbiog.mwx`) and the leaflet files of all modules. Do the same with `mw_phys.ini` but retain the full MWX file complement only for the M2TFNS, E1CPLX and E4VECT modules.

- You need to add appropriate descriptive text and file references for the exam files in the [autoindex] section of d:\mw\mw.ini. For the sociologists their exam files and the reference to the HTML file should be added to the [autoindex] section of mw_soc.ini, whilst the physicists need only the course notes text file added to the same section of mw_phys.ini. For each of the three .INI files useAutoIndex should be set to 1, addToPalette set to 2 and name to University X.
- Both the **Mathwise** setup (using mw.ini) and the **Sociology** setup (using mw_soc.ini) now include non-standard files: Microsoft Word files in both cases and also HTML for the sociologists. We need to inform Mathwise how to deal with these files (see **Chapter 5**). For both .INI files you should add the Word reader application at f:\apps\mw\word\winword.exe to the [associations] section, associating it with the doc suffix. You should also add in mw_soc.ini a reference to Netscape (at f:\apps\browsers\netscape\netscape.exe) and associate it with the htm suffix. No changes to the [associations] section of mw_phys.ini are necessary.
- Finally we need to make some changes to the entry unit to make sure that the material we added is available from the start of a Mathwise session (see **Chapter 6**). Open entryu.ini, the default initialisation file for the entry unit. Both the **Mathwise** and the **Sociology** setups will use this file. In the [special] section of entryu.ini set useSpecial to 1, buttonName to University X and add one descriptor query: desc1=UTIL,AUTO. Close and save entryu.ini, mw.ini and mw_soc.ini.
- The **Physics** setup is a little different. Make a copy of the newly modified entryu.ini and call it eu_phys.ini. Leave the [special] section as it is but also add a second and third descriptor query to list learning units and leaflets that deal with complex numbers i.e., add the lines desc2=UNIT,,,complex and desc3=LEAF,,,trig. (Note that you could just enter this as one descriptor using a boolean search i.e., desc2=LEAF,,,complex or trig). As you are overriding the default entry unit initialisation you need to make a final modification to mw_phys.ini. In the [general] section of the file change startupDesc=EUNT,, to startupDesc=EUNT,,,_c:\mw\eu_phys.ini. Close and save eu_phys.ini and mw_phys.ini. You have now completed the customisation and should run all three setups to test your work.

The Solution

Just to clarify the previous steps what follows are extracts of the relevant parts of the various files you have edited.

mw.ini : added auto-index material and association for Word

```

...
[autoindex]
useAutoIndex=1
addToPalette=2
name=University X
Exam 1996=c:\support\maths\mexam11.doc
Exam 1997=c:\support\maths\mexam12.doc
...
[associations]
Assoc1Name=Toolbook
Assoc1Ext=tbk
Assoc1Path=tbook.exe
...
Assoc12Name=Microsoft Word
Assoc12Ext=doc
Assoc12Path=f:\apps\mw\word\winword.exe

```

```

entryu.ini : modified [special] section
...
[special]
useSpecial=1
buttonName=University X
desc1=UTIL,AUTO
...

mw_soc.ini : modified module list, added auto-index material and added associations
...
[modules]
EntryUnit=ent_unit.mwx
blbiog=e:\modules\blbiog\blbiog.mwx
mwtools=mwtools.mwx

mlgrph=e:\modules\mlgrph\mlgrph.mwx
mlgrphLeafs=e:\modules\mlgrph\mlgrph_1.mwx

mlbase=e:\modules\mlbase\mlbase.mwx
mlbaseLeafs=e:\modules\mlbase\leaflets\mlbase_1.mwx
mlbaseLU1=e:\modules\mlbase\mlbase1\mlbase1.mwx
mlbaseLU2=e:\modules\mlbase\mlbase2\mlbase2.mwx
mlbaseLU3=e:\modules\mlbase\mlbase3\mlbase3.mwx
mlbaseLU4=e:\modules\mlbase\mlbase4\mlbase4.mwx
mlbaseLU5=e:\modules\mlbase\mlbase5\mlbase5.mwx
mlbaseTest=e:\modules\mlbase\mlbaset\mlbaset.mwx

mlineqLeafs=e:\modules\mlineq\mlineq_1.mwx
mllogsLeafs=e:\modules\mllogs\mllogs_1.mwx
m2tfnsLeafs=e:\modules\m2tfns\m2tfns_1.mwx
elcplxLeafs=e:\modules\elcplx\elcplx_1.mwx
e4vectLeafs=e:\modules\e4vect\e4vect_1.mwx
...
[autoindex]
useAutoIndex=1
addToPalette=2
name=University X
Sample paper 1=c:\support\soc\s_exam1.doc
Sample paper 2=c:\support\soc\s_exam2.doc
Sociology Web page=s:\www\soc\start.htm
...
[associations]
Assoc1Name=Toolbook
Assoc1Ext=tbk
Assoc1Path=tbook.exe
...
Assoc12Name=Microsoft Word
Assoc12Ext=doc
Assoc12Path=f:\apps\mw\word\winword.exe
Assoc12Name=HTML
Assoc12Ext=htm
Assoc12Path=f:\apps\browsers\netscape\netscape.exe

```

mw_phys.ini : over-ride default entry unit, modified module list and added auto-index material

```
[general]
...
startupDesc=EUNT,,,,_c:\mw\eu_phys.ini
...
[modules]
EntryUnit=ent_unit.mwx
blbiog=e:\modules\blbiog\blbiog.mwx
mwtools=mwtools.mwx

mlgrphLeafs=e:\modules\mlgrph\mlgrph_l.mwx
mlineqLeafs=e:\modules\mlineq\mlineq_l.mwx
mlbaseLeafs=e:\modules\mlbase\leaflets\mlbase_l.mwx
mllogsLeafs=e:\modules\mllogs\mllogs_l.mwx

m2tfns=e:\modules\m2tfns\m2tfns.mwx
m2tfnsLeafs=e:\modules\m2tfns\m2tfns_l.mwx
m2tfnsTest=e:\modules\m2tfns\m2tfnst\m2tfnst.mwx

elcplx=e:\modules\elcplx\elcplx.mwx
elcplxLeafs=e:\modules\elcplx\elcplx_l.mwx
elcplxTest=e:\modules\elcplx\elcplx_t\elcplx_t.mwx

e4vect=e:\modules\e4vect\e4vect.mwx
e4vectLeafs=e:\modules\e4vect\e4vect_l.mwx
e4vectTest=e:\modules\e4vect\e4vect_t\e4vect_t.mwx

[autoindex]
useAutoIndex=1
addToPalette=2
name=University X
Course notes= c:\support\physics\cnotes1.txt
...
```

eu_phys.ini : modified [special] section

```
...
[special]
useSpecial=1
buttonName=University X
desc1=UTIL,AUTO
desc2=LEAF,,,,complex
desc3=LEAF,,,,trig
...
```

Appendix 1

Long File names

Mathwise in its current version (v2.1) is designed to be used with all versions of Microsoft Windows, including Windows 3.1. One of the main differences between Windows 3.1 and later is that file names were limited to 8 characters maximum, plus 3 for the file suffix in the earlier operating system. From Windows 95 (32-bit versions) onwards the operating system has supported long file and directory names, these can be of more or less any length and format. By its nature Mathwise acts as a kind of extension to the Windows file system, and hence relies heavily on file-names. Hence, this change to long file names can produce some problems. To retain compatibility with Windows 3.1, Mathwise uses the old, limited file name system (8 + 3 characters maximum) and this must be considered when installing and customising Mathwise on newer Windows systems.

Microsoft has recognised the difficulties in the new file name system for Windows 3.1 compatible software and provides a method of dealing with it. You should understand how this works if you are using Mathwise in a situation where long file names are in use. In particular this applies when you are editing initialisation files to customise Mathwise where files you refer to (which might perhaps be auto-indexed files or new reader applications) are located and/or named with long file names.

The method, in principle, is to refer to files and directories that have long file names with a compressed version of the file name. The compression works by taking the first six characters of the long file name then adding the ~ symbol, followed by a digit. The digit is used to distinguish between different files with the same first six characters. Note that long file names that would be acceptable under the old system remain as they are. Here are some examples.

Long file name	becomes	Short file name
ref1.doc		ref1.doc
reference1.doc		refere~1.doc
reference1.txt		refere~1.txt
referencebook.txt		refere~2.txt

In addition, since long file names can also contain spaces, (which cannot be allowed in the old system), spaces are removed and the ~ and digit are added. Note that this occurs even for names that are less than 8 + 3 characters long before the removal of the spaces. e.g.

Long file name	becomes	Short file name
r book.txt		rbook~1.txt
ref book.txt		refboo~1.txt

In cases where the file suffix is greater than three characters it too is shortened to 3 characters and the 6 characters plus ~ plus digit effect is applied to the prefix part of the file name, e.g.

Long file name	becomes	Short file name
screen1.htm		screen1.htm
screen1.html		screen~1.htm
screen10.html		screen~2.htm

The same idea applies to directories — if unacceptable under the old system the directory name is ‘de-spaced’, shortened to the first six characters and then appended with ~ and digit, e.g.

Long file name	becomes	Short file name
C:\docs\files\ C:\docs\mydatafiles\ C:\docs\mydatafiling\ C:\docs\my files\ 		C:\docs\files\ C:\docs\mydata~1\ C:\docs\mydata~2\ C:\docs\myfile~1\

One of the best ways to see this system in action is to open an MS-DOS session in Windows 95 and use the `dir` command to view directories that containing long file names then compare the MS-DOS

rendering of the names with that displayed in Explorer for example. It is essential to do this if you are adding long file named files to Mathwise, since the digit after the ~ is allocated by Windows and in cases where it rises above 1 the allocation is impossible to predict. In the example above mydatafile\ might reduce to mydata~2 rather than mydata~1; there is no way of knowing without looking.

This is the principle of dealing with long-file names within Mathwise. The practice, unfortunately, is sometimes more difficult. First we have found that in some cases, with no discernible pattern, the substituted file or directory names simply do not work. If you find this to be the case the only solution is to change file and directory names to the old style. Secondly the shortenings and ~ plus digit allocations are not at all easy to predict. The safest approach is therefore to ascertain them using an MS-DOS session, using the **File Open** dialog of a Windows 3.1 application or perhaps best of all the old Windows 3.1 File Manager (which can be found in the Windows 95 Windows directory as winfile.exe).

For further information on dealing with long file names you should consult the Mathwise website at <http://www.bham.ac.uk/mathwise>.