



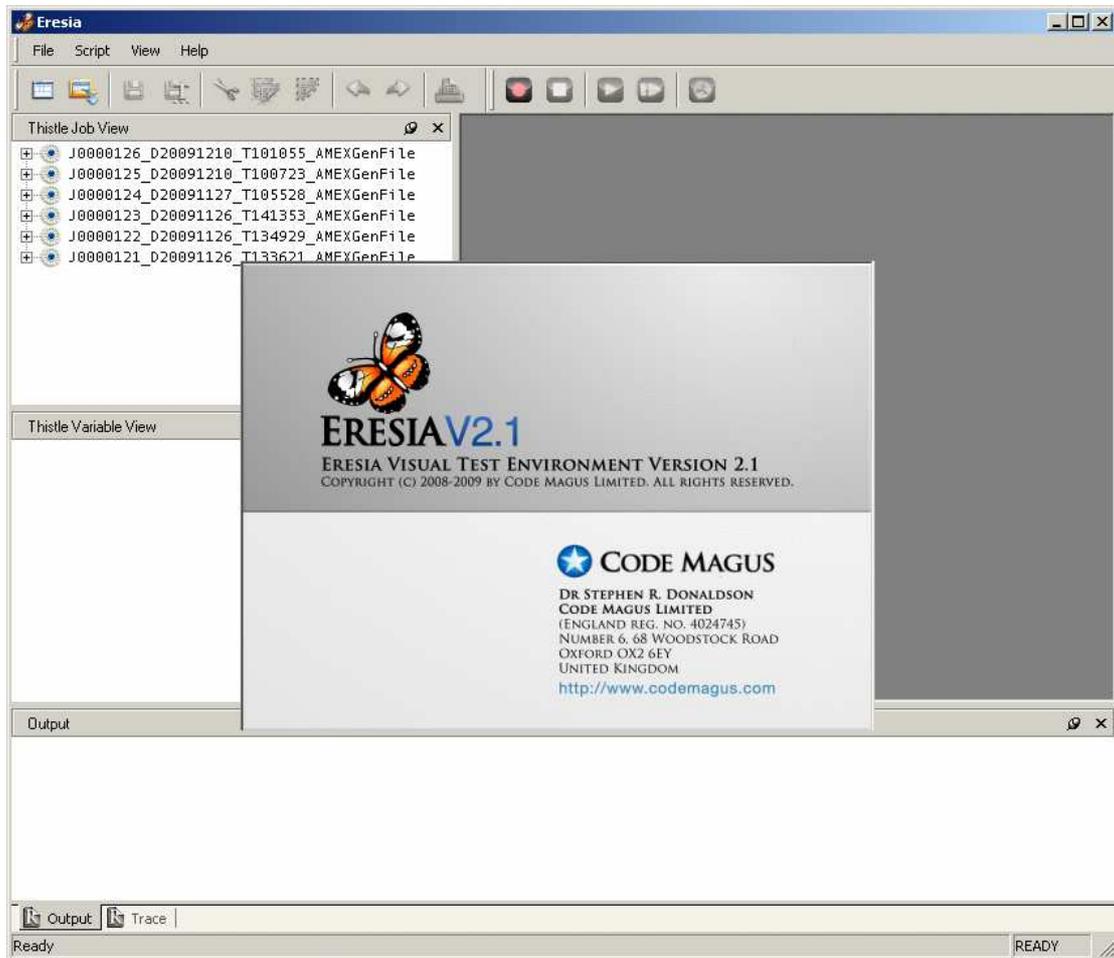
Code Magus Eresia User Guide Version 2.1

CML00040-21

Code Magus Limited (England reg. no. 4024745)
Number 6, 69 Woodstock Road
Oxford, OX2 6EY, United Kingdom
www.codemagus.com
Copyright © 2009 Code Magus Limited
All rights reserved



January 21, 2010



Contents

1	Introduction	4
1.1	Overview	4
1.2	Summary	4
2	The Graphical User Interface	5
2.1	Output Windows	5
2.1.1	Thistle Job View Window	5
2.1.2	Thistle Variable View Window	7
2.1.3	Eresia Output Window	8
2.1.4	Script View Window	9
2.2	Menu Items	10
2.2.1	File	10
2.2.2	Debug	11
2.2.3	Edit	12
2.2.4	Search	13
2.2.5	Window	14
2.2.6	Help	14
2.3	The Toolbar	16
2.3.1	New Script	16
2.3.2	Playback	16
2.3.3	Find and Replace	16
2.3.4	Debug	17
2.3.5	Bookmarks	17
2.4	Dialogs	18
2.4.1	Preferences	19
2.4.2	Thistle Config Editor	20
2.4.3	Find	20
2.4.4	Replace	21
2.4.5	Record	21
2.5	Help System	22
2.5.1	Contents and Index	22
2.5.2	Thistle Help	23
2.5.3	Object Type Help	24
2.5.4	About Eresia	25
3	Using the Visual Test Environment	26
3.1	Opening a script	26
3.2	Setting breakpoints	27
3.3	Running Eresia scripts	28
3.3.1	Running Eresia in non-debug mode	28
3.3.2	Running Eresia debug mode	30
3.4	Viewing Thistle variables during execution	30

CONTENTS 3

4 Example Scripts 33

1 Introduction

1.1 Overview

The Eresia Visual Test Environment (VTE) is a Graphical User Interface (GUI) whose primary function is to allow users to view, edit and execute Eresia scripts.

The VTE can also be used to record scripts from other portals, such as the the Network Injection Portal or File Injection Portal among others.

During recording these portals generate real scripts in the VTE which can then be used from within the VTE without having to run them in the portal.

There are additional portals which do not have their own GUI, such as the Type A portal, the Excel portal, and others. These will be discussed later in the document. See [Section 3](#) on page [26](#).

The GUI is divided into various sections. These sections are:

- A log of the activity occurring during the running of scripts
- A set of associated files and output associated with a job
- An area showing the actual scripts
- An output area where information is printed during script execution

These windows are explained in detail in section [Subsection 2.1](#) on page [5](#).

1.2 Summary

The VTE is the control center for all of the Eresia tools. From within the VTE, test scripts are executed in the form of Eresia scripts.

These scripts can access the other Eresia and Code Magus tools by using portals which each have a specific function.

The content of these scripts can be generated by recording from within one of the other Code Magus GUI tools such as the NIP, FIP, XIP and 3270 Portal, amongst others.

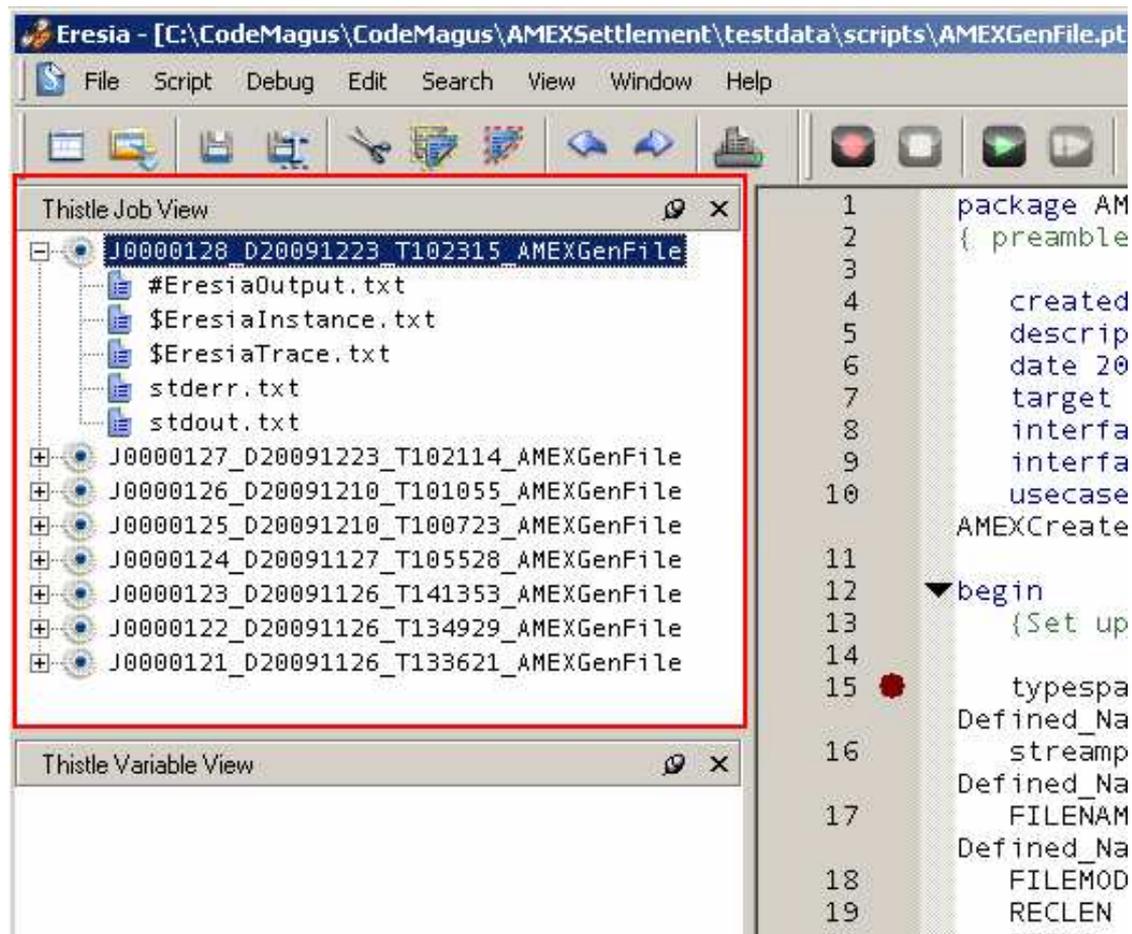
For each execution of a package, a log of the activity is recorded to the GUI so the user can track the progress of the job. A job is an instance of a Package execution. The job contains the logs and files that the script accesses. This allows the user to inspect the data that was manipulated during the execution of the job. The job information is stored on disk. This means that audit and trace information is stored on disk for future reference.

2 The Graphical User Interface

This section details all of the components and visual artifacts that make up the GUI.

2.1 Output Windows

2.1.1 Thistle Job View Window



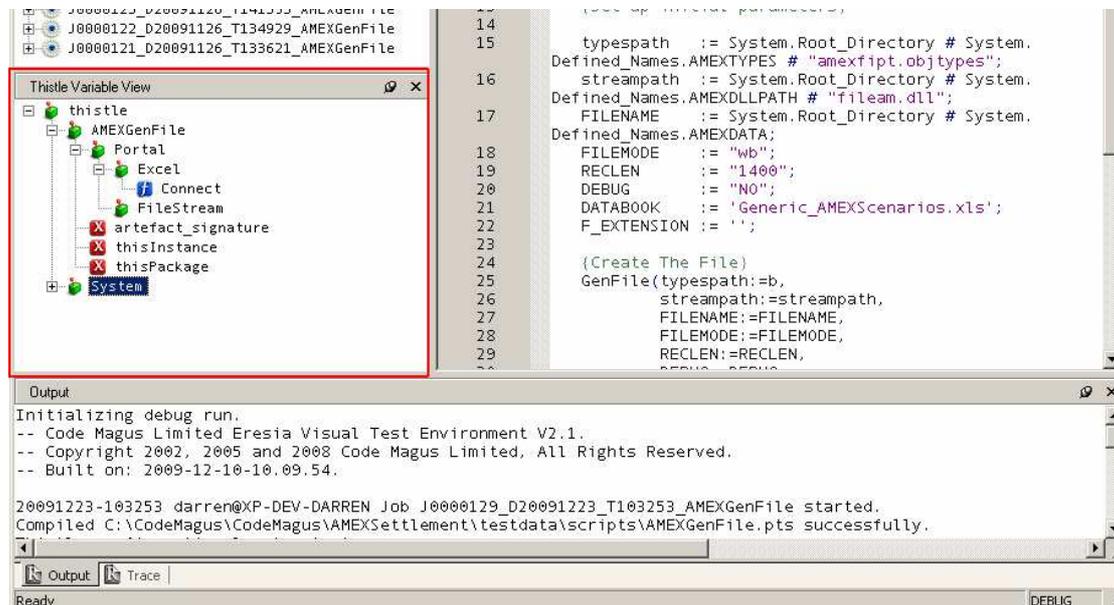
The red high-lighted window a history of all of the jobs that have been executed, and each entry contains a log of the activity and the associated files used during the execution of that job. When the top-level job node is expanded, a list of the associated input and output of the job can be explored and selected.

Each of the top-level nodes (a job) represent an instance of a package execution. The entries under the top node refer to documents used in and output that was generated by the execution of the package.

Note that the script itself can write to a file in the console. This is also true of any of the

Eresia portals and interfaces.

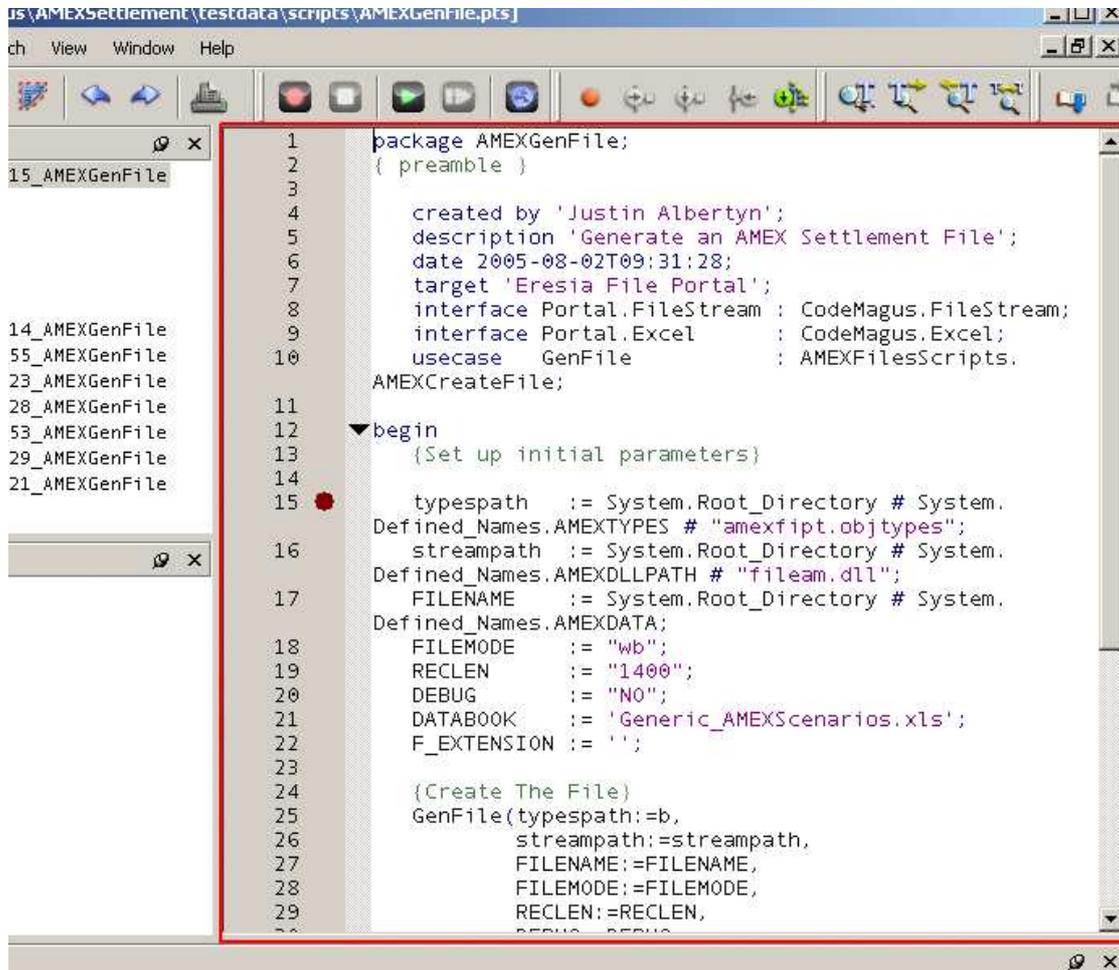
2.1.2 Thistle Variable View Window



The red high-lighted window shows the entire variable tree of the current package.

This tree shows a hierarchical view of all of the variables in the current instance. These variables can be altered at run-time (for example, when a breakpoint has been set, or the VTE is waiting for user feedback).

2.1.4 Script View Window



The red high-lighted window is where Eresia scripts are displayed. This is also the text are that is the subject of the editor functions.

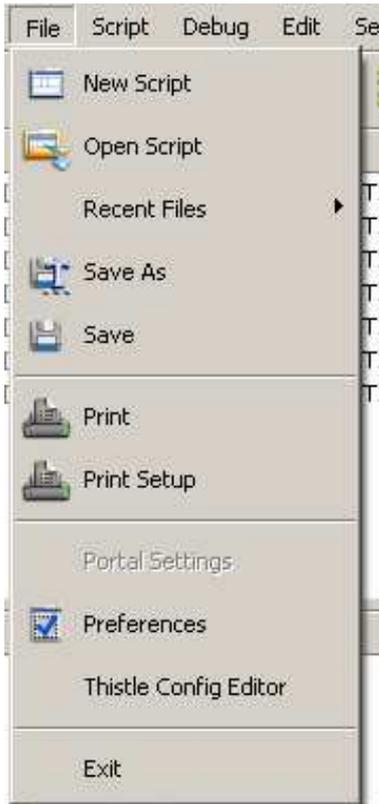
The user can use Find/Replace and Copy/Cut/Paste functions here.

Breakpoints can be set where the debugger will stop (if the user runs the script in debug mode).

2.2 Menu Items

This section shows the menu options in the VTE.

2.2.1 File

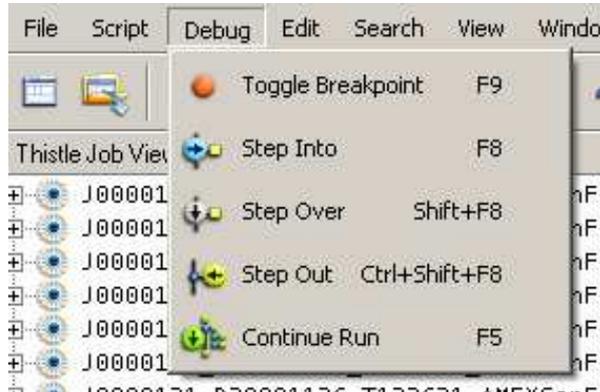


Menu items	
New Script	Create a new, empty script
Open Script	Open an existing script file
Recent Files	Shows a list of recently used files
Save As	Save the current script under a different filename
Save	Overwrite the current script file with the new script
Print	Print the current script
Print Setup	Setup the printing preferences
Preferences	Open the preferences dialog
Thistle Config Editor	Open the Thistle Config Editor dialog
Exit	Close Eresia Visual Test Environment

- *New Script*: Create a new script in the script editor window. This empty script will later be saved to disk and opened in the editor

- *Open Script*: Open an existing Eresia script from a file on disk. This script can then be edited, or breakpoints can be set for debugging during script execution
- *Recent Files*: Displays a list of all of the scripts that have been opened recently. The user can then select the file to open it in the editor window
- *Save As*: Save the script under a different filename. This can be used to make copies of files that may need only small adjustments, or as templates
- *Save*: Save any changes made to the script. This will overwrite the existing file
- *Print*: Send the current script to a printer
- *Preferences*: Brings up the preferences window to change Eresia settings such as the font, font size and editor styles amongst others
- *Thistle Config Editor*: Brings up the thistle config window where variables can be set up for the current package
- *Exit*: Close and exit from Eresia. The user will be prompted to save their changes if the changes have not yet been saved

2.2.2 Debug

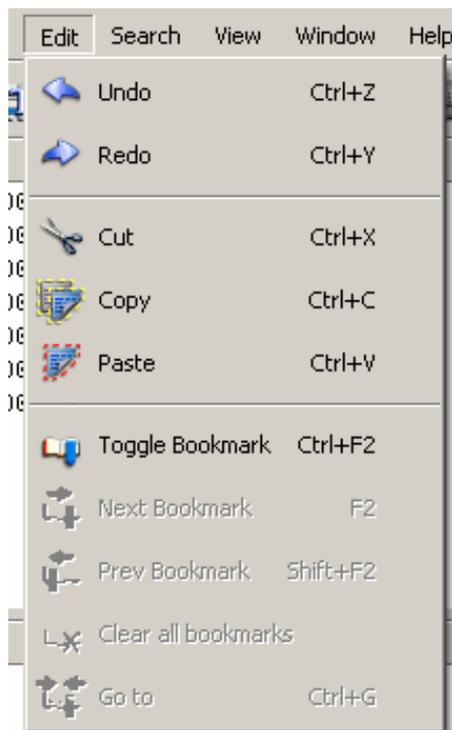


Menu items	
Record	Record a new script
Stop	Stop the recording
Run	Run the current package script
Continue	Resume the current script
Compile	Compile the current script

- *Record*: Brings up the Eresia Record window to select the portal and other options for recording. A new script window is also opened where the recorded script will be generated
- *Stop*: End the recording or playback process and return to Eresia

- *Run*: Run the currently selected script. This will bring up the required portal
- *Continue*: Resume the playback of the current execution, for example when Eresia is currently stopped at a breakpoint
- *Compile*: Compile the current script. This is used to check for errors in the script

2.2.3 Edit

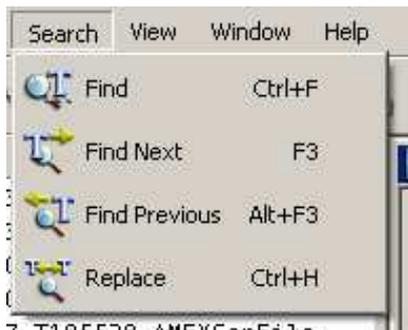


Menu items	
Undo	Undo the last action
Redo	Redo the last action
Cut	Cut the current selected text
Copy	Copy the current selected text
Paste	Paste the current clipboard text
Toggle Bookmark	Set a bookmark
Next Bookmark	Go to the next bookmark
Prev Bookmark	Go to the previous bookmark
Clear all bookmarks	Reset all bookmarks
Go To	Go to a bookmark

- *Undo*: Cancel changes that have been made in the script. This can be cancelled by selecting Redo
- *Redo*: Redo changes that have been undone with the Undo command

- *Cut*: Cut the currently selected text from the script, and place it in the clipboard
- *Copy*: Copy the currently selected text in the script into the clipboard
- *Paste*: Paste the text currently in the clipboard into the script at the position of the cursor
- *Toggle Bookmark*: Set a bookmark in a script. This is used to mark lines of interest in a script but do not break execution like a breakpoint does
- *Next Bookmark*: Move the cursor to the next bookmark
- *Prev Bookmark*: Move the cursor to the previous bookmark
- *Clear all bookmarks*: Remove all bookmarks
- *Go To*: Go to a specific bookmark

2.2.4 Search



Menu items	
Find	Open the Find Dialog to search text
Find Next	Go to the next match
Find Previous	Go to the prev match
Replace	Open the Replace Dialog

- *Find*: Search for text in the current script
- *Find Next*: Find the next matching text set by using the Find command
- *Find Previous*: Find the previous matching text set by using the Find command
- *Replace*: Replace an expression or text with another expression or text

2.2.5 Window



Menu items	
Tile Horizontal	Tile open scripts horizontally
Tile Vertical	Tile open scripts vertically
Cascade Windows	Cascade all open scripts

- *Tile Horizontal*: Arrange the open script windows in equally spaced horizontal strips
- *Tile Vertical*: Arrange the open script windows in equally spaced vertical strips
- *Cascade Windows*: Arrange the open script windows so that they are cascaded on top of each other

2.2.6 Help



Menu items	
Contents and Index	Open the VTE help file
Thistle Help	Open the Thistle help file
Object Types Help	Open the Object Types help file
About Eresia	Open the About Dialog

- *Contents and Index*: Show the Eresia help pages

- *Thistle Help*: Show the Thistle Help manual. This manual describes the Thistle language
- *Object Types Help*: Show the Object Types manual. This manual describes the Object Types library used to manipulate Object Types
- *About Eresia*: Show the About Eresia window which displays version and date information, and also displays contact details for Code Magus Ltd

2.3 The Toolbar

This section describes the toolbar in Eresia and what the icons mean.



2.3.1 New Script



Open a new script in the text editor.

2.3.2 Playback



The icons in this section in order are:

Menu items	
Record	Record a new script
Stop	Stop the recording
Play	Run the current package script
Continue	Resume the current script
Compile	Compile the current script

- *Record*: Brings up the Eresia Record window to select the portal and other options for recording. A new script window is also opened where the recorded script will be generated
- *Stop*: End the recording or playback process and return to Eresia
- *Run*: Run the currently selected script. This will bring up the required portal
- *Continue*: Resume the playback of the current execution, for example when Eresia is currently stopped at a breakpoint
- *Compile*: Compile the current script. This is used to check for errors in the script

2.3.3 Find and Replace



The icons in this section in order are:

Menu items	
Find	Open the Find Dialog to search text
Find Next	Go to the next match
Find Previous	Go to the prev match
Replace	Open the Replace Dialog

- *Find*: Search for text in the current script
- *Find Next*: Find the next matching text set by using the Find command
- *Find Previous*: Find the previous matching text set by using the Find command
- *Replace*: Replace an expression or text with another expression or text

2.3.4 Debug



The icons in this section in order are:

Menu items	
Set Breakpoint	Set a breakpoint on the current line
Step Into	Step Into at the current line
Step Over	Step Over at the current line
Step Out	Step Out at the current line
Continue run	Continue running script normally

- *Set Breakpoint*: Set a breakpoint at the cursor. This will cause the debugger to break execution when this line is reached in the script
- *Step Into*: Step into the current execution block. This will force the debugger to open the function that the code points to (for example)
- *Step Over*: Step over the current execution block. This will force the debugger to pass over the script that the line points to
- *Step Out*: Finish execution of the current block and step out of the current scope
- *Continue run*: Resume running the script normally

2.3.5 Bookmarks



The icons in this section in order are:

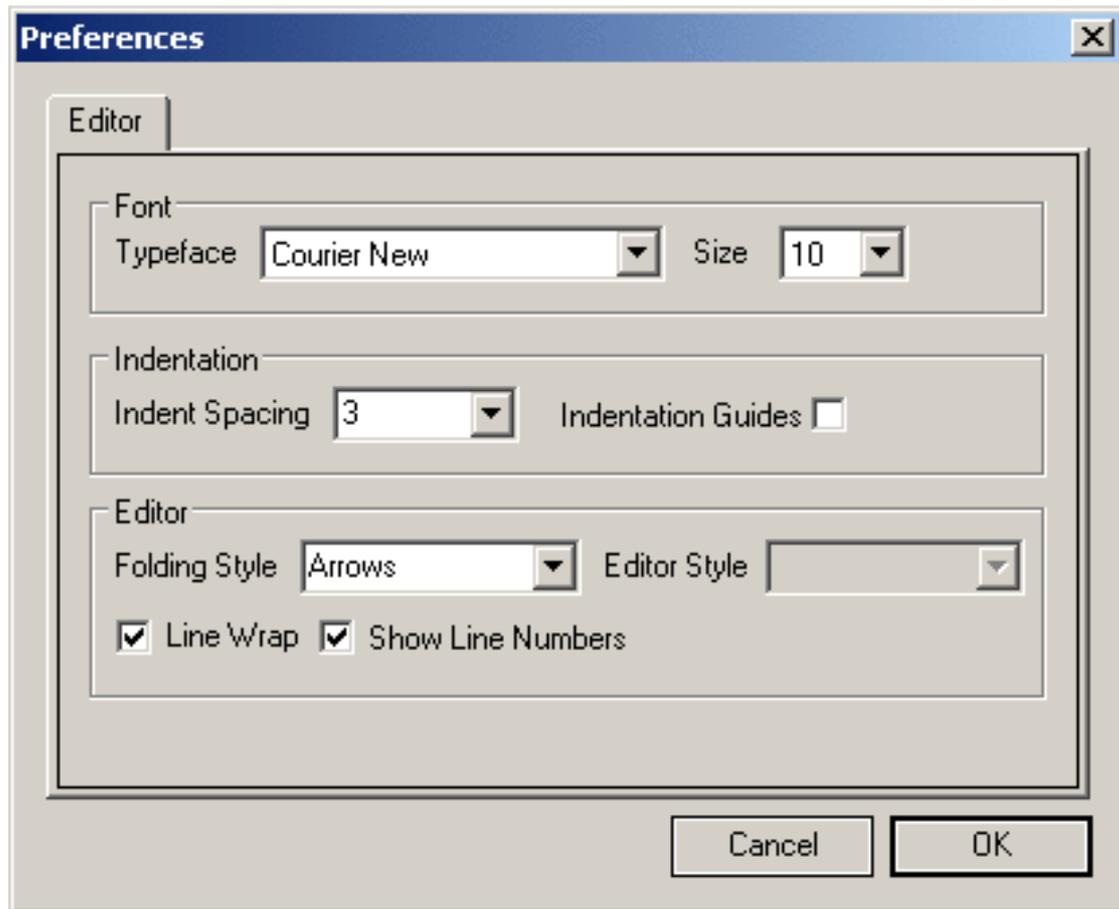
Menu items	
Toggle Bookmark	Set a bookmark
Next Bookmark	Go to the next bookmark
Prev Bookmark	Go to the previous bookmark
Clear all bookmarks	Reset all bookmarks
Go To	Go to a bookmark

- *Toggle Bookmark*: Set a bookmark in a script. This is used to mark lines of interest in a script but do not break execution like a breakpoint does
- *Next Bookmark*: Move the cursor to the next bookmark
- *Prev Bookmark*: Move the cursor to the previous bookmark
- *Clear all bookmarks*: Remove all bookmarks

2.4 Dialogs

This section describes all of the dialogs used for configuration and setup of the environment.

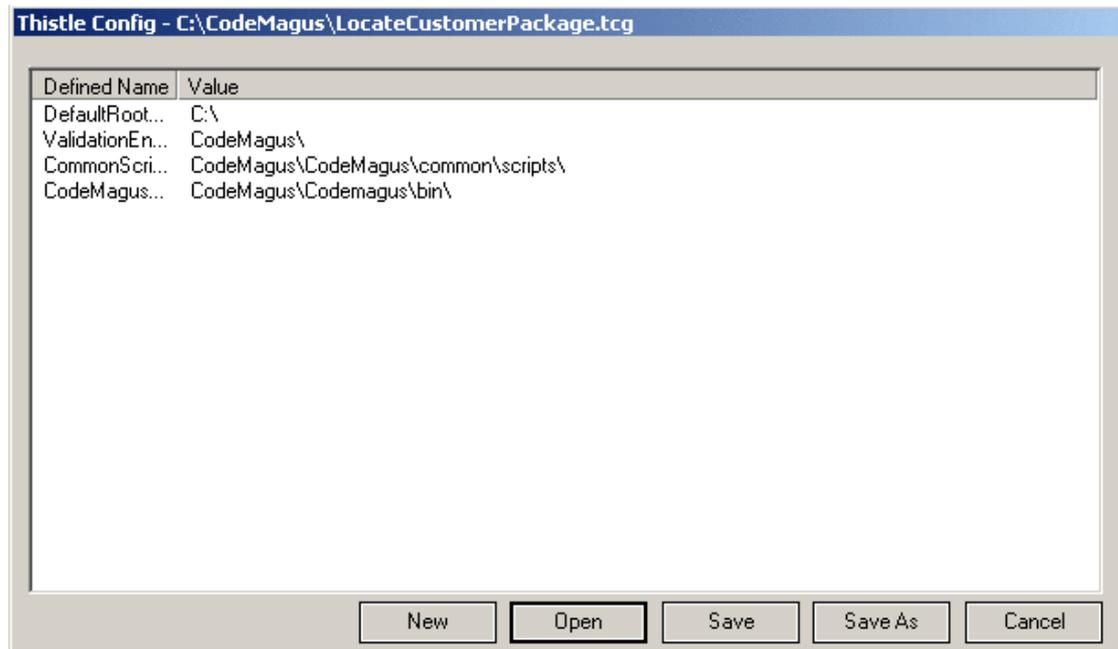
2.4.1 Preferences



This dialog allows the user to specify:

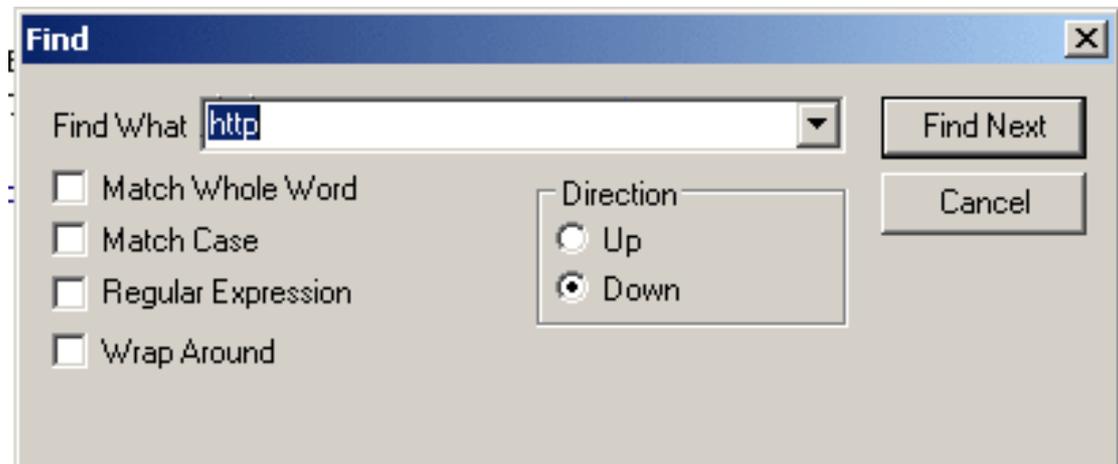
- The font size and family for the script editor
- The indentation width for tabs
- The style of the folding arrows of the script editor
- Toggling of line numbers in the script editor
- Toggling of line wrapping in the script editor

2.4.2 Thistle Config Editor



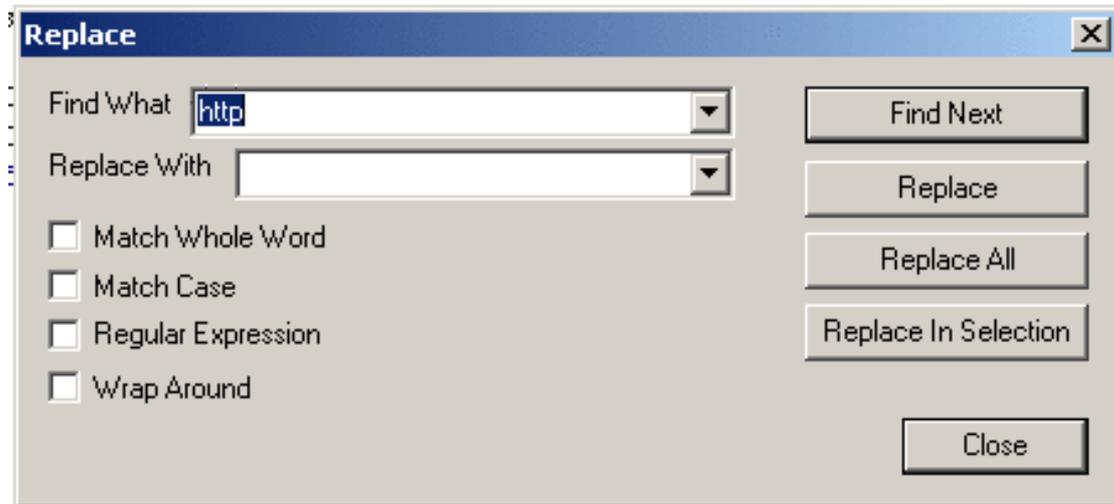
This dialog allows the user to enter defined names for use in the VTE.

2.4.3 Find



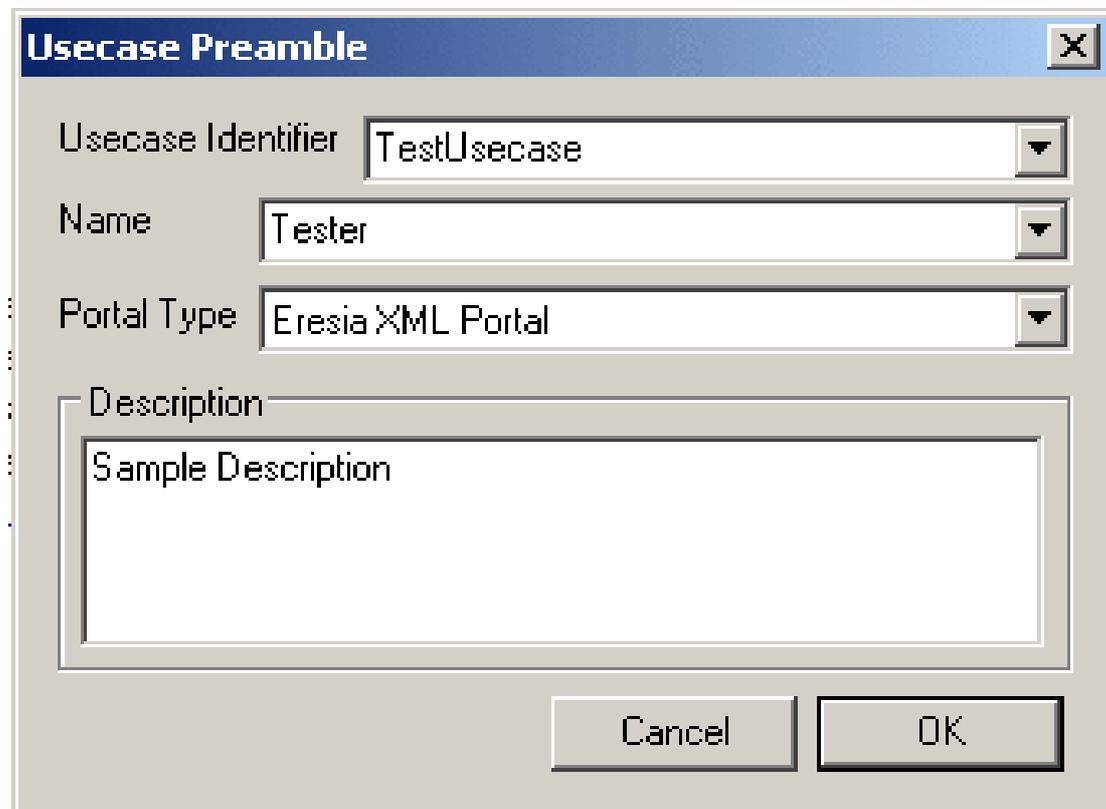
This dialog allows the user search the current script for text using regular expressions.

2.4.4 Replace



This dialog allows the user search the current script for text using regular expressions, and then replace any matches.

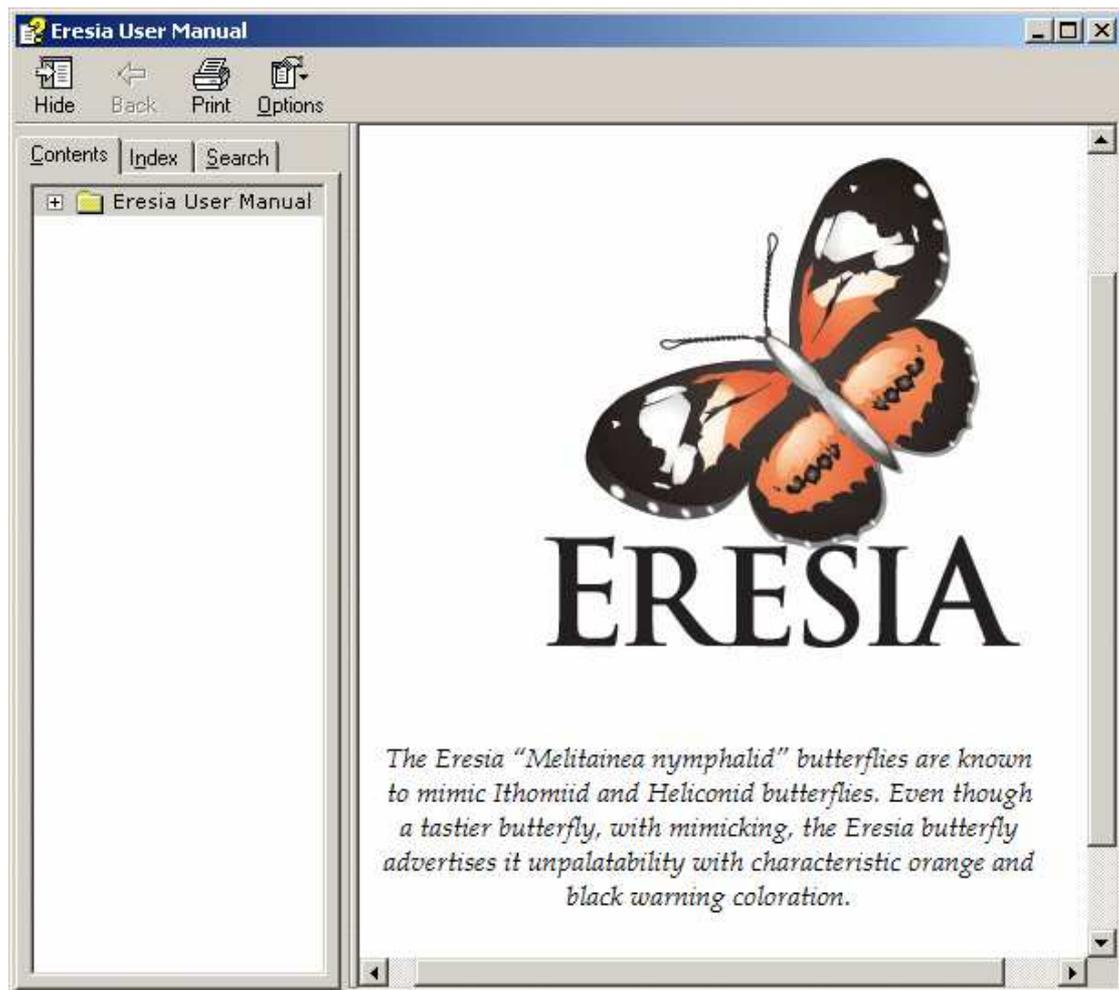
2.4.5 Record



This dialog brings up the dialog to begin recording in another portal such as the XIP, FIP, NIP or 3270 portals.

2.5 Help System

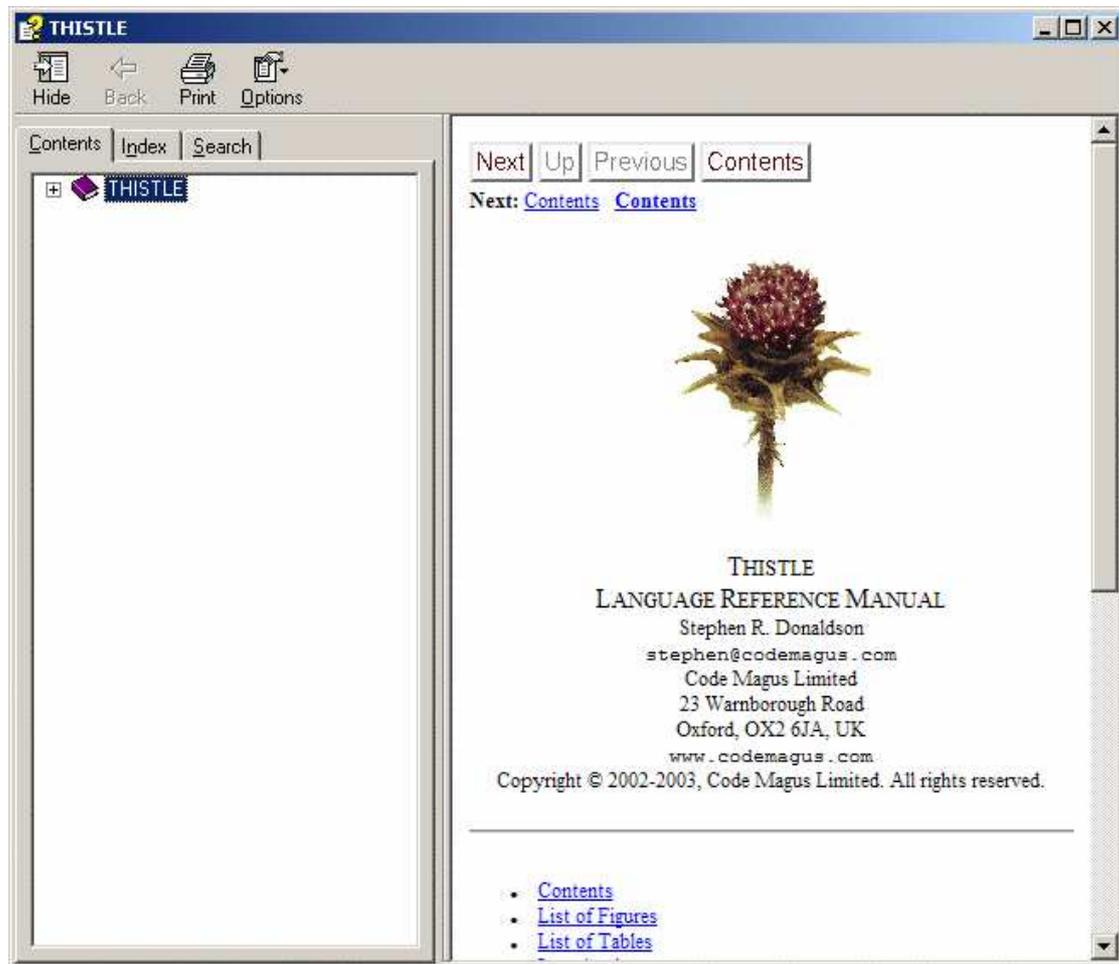
2.5.1 Contents and Index



This is the Help for the Visual Test Environment. This help relates to the actual usage of the VTE, and includes examples of how to record and playback scripts.

This window can be accessed by selecting Help, then Contents and Index from the menu bar.

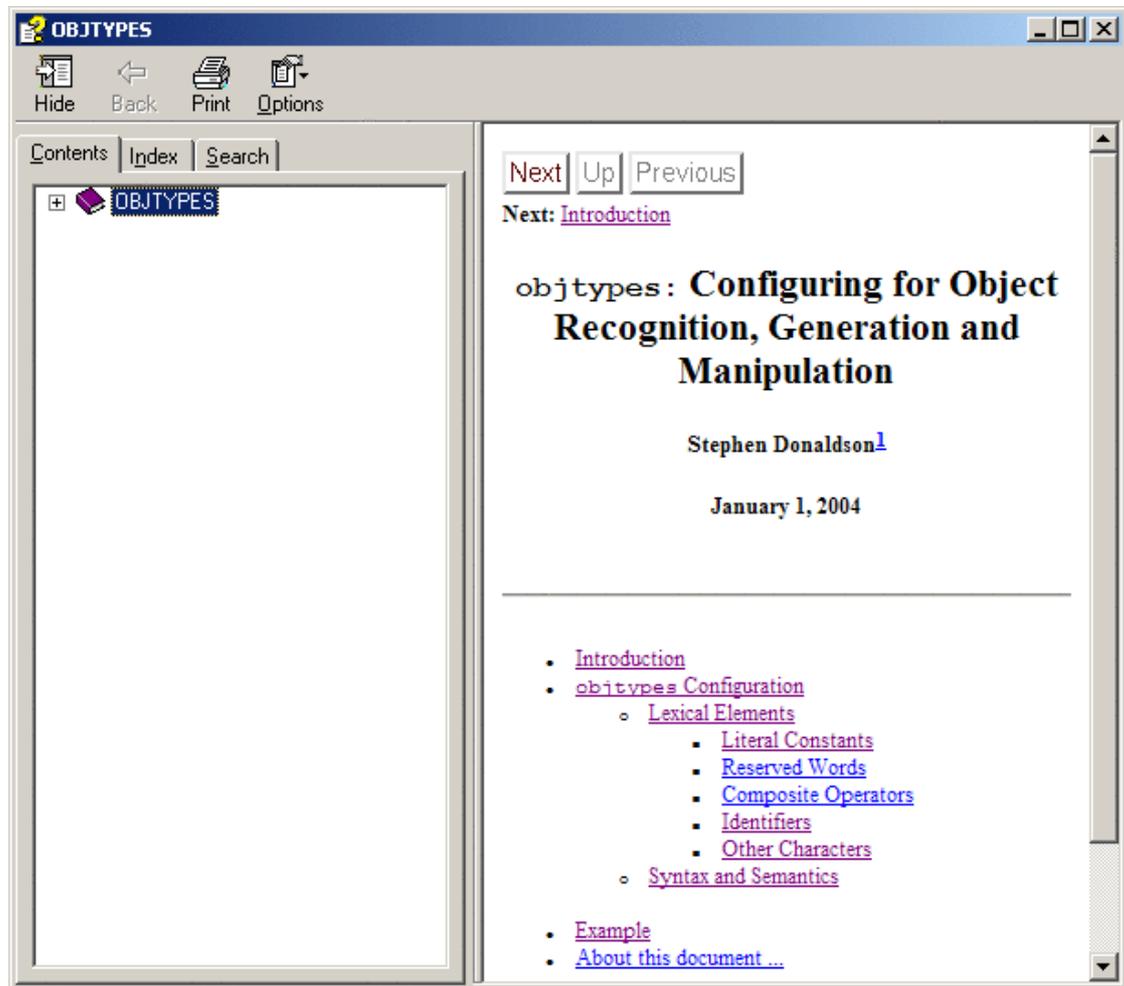
2.5.2 Thistle Help



This displays the Thistle help document, which explains the usage of Thistle and includes the list of reserved words and other examples of using the Thistle library.

This window can be accessed by selecting Help, then Thistle Help from the menu bar.

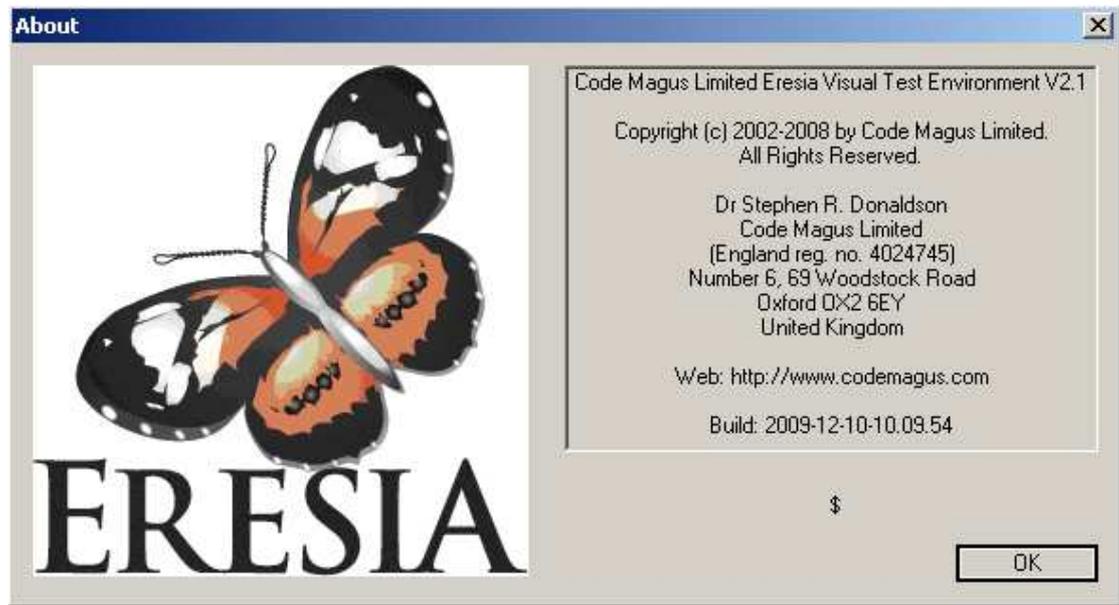
2.5.3 Object Type Help



This displays documentation from Object Types, and explains how to use the object types library to identify messages. It includes examples of scripts that use object types, and also information about the syntax and usage of the library.

This window can be accessed by selecting Help, then Object Types Help from the menu bar.

2.5.4 About Eresia



The Help About dialog shows information about the Visual Test Environment. It also shows the current build of the software.

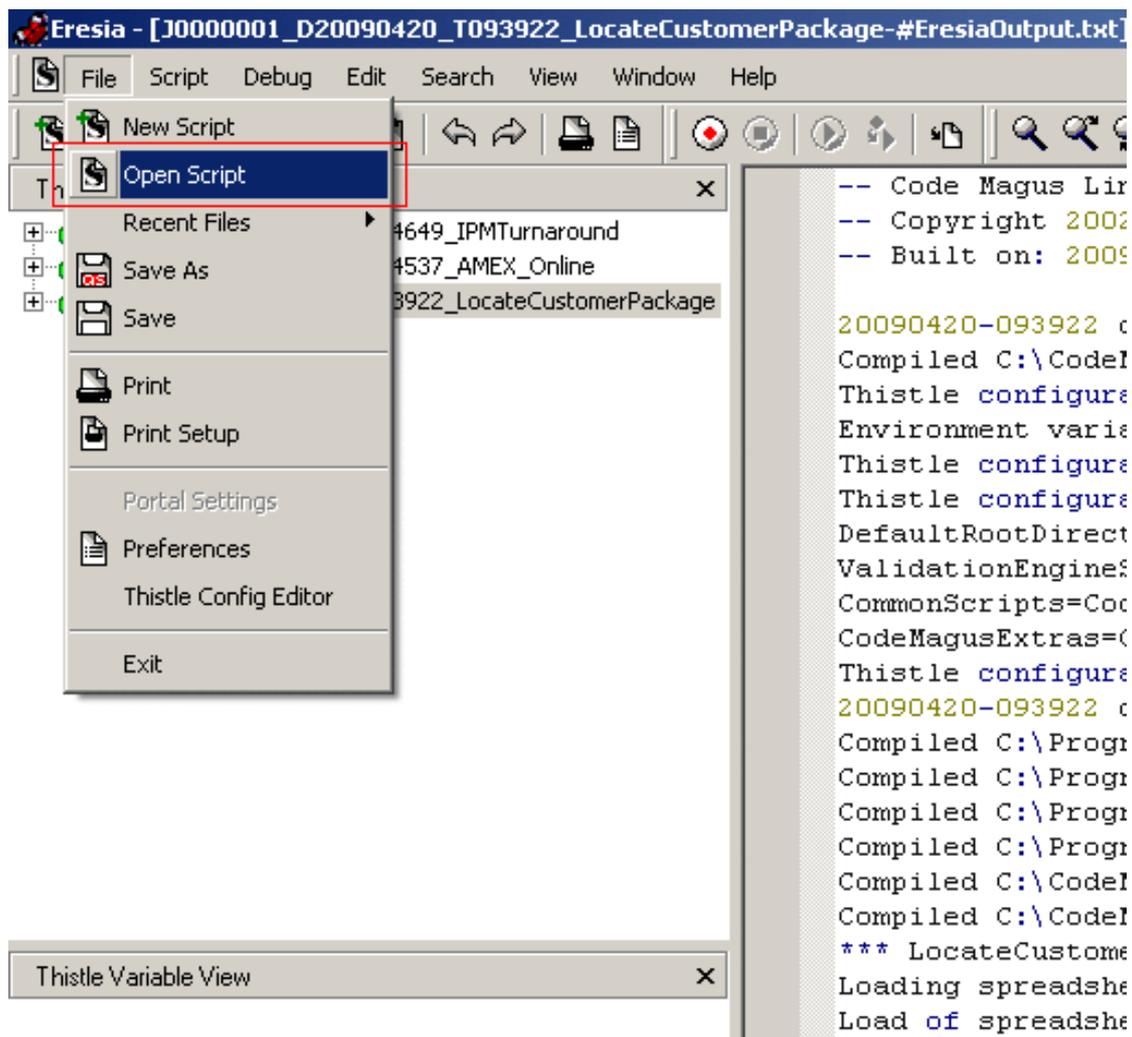
This window can be accessed by selecting Help, then About Eresia from the menu bar.

3 Using the Visual Test Environment

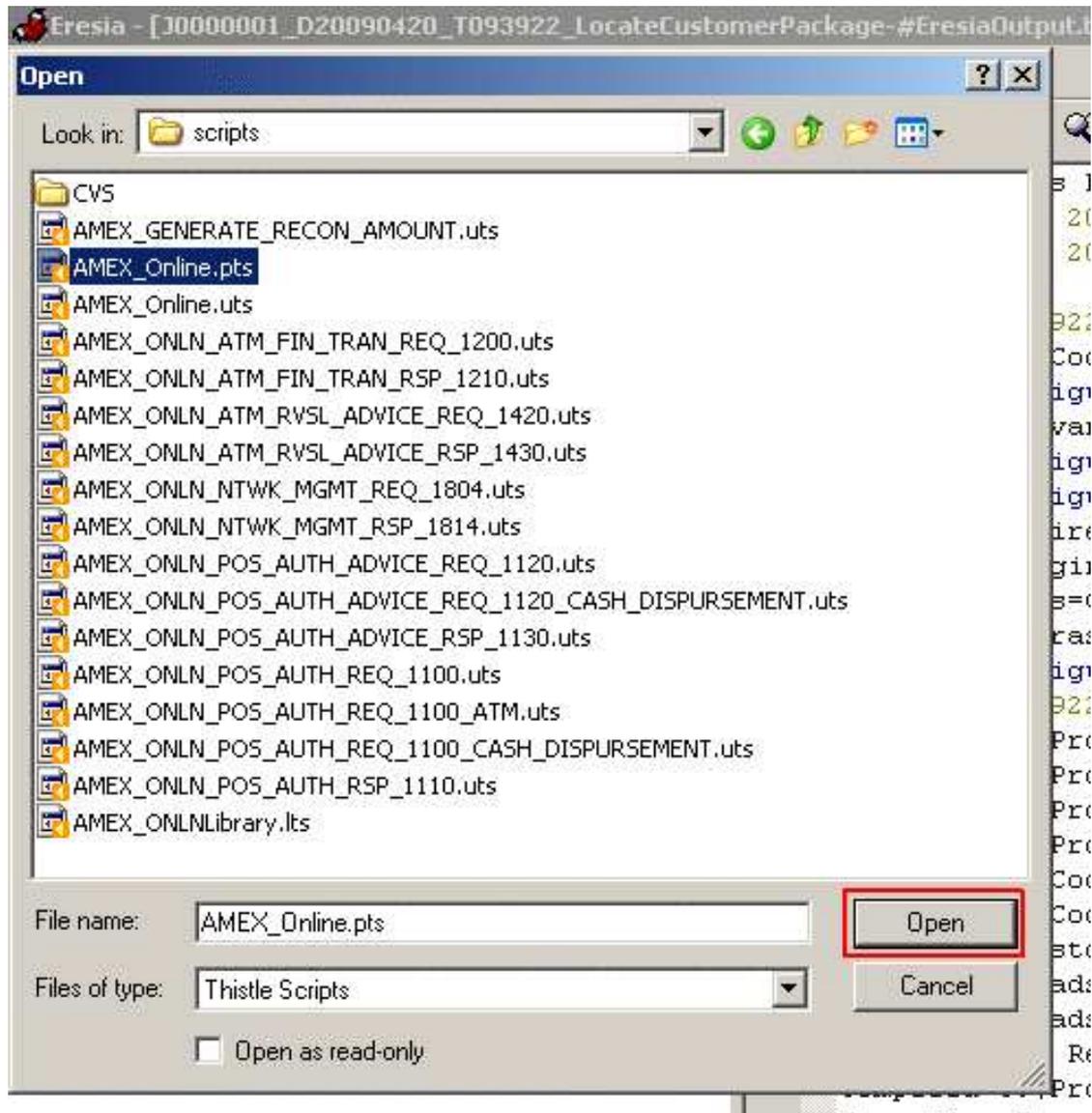
3.1 Opening a script

Scripts can be opened in the GUI by using the File menu, by using the Recent Files menu, or by double-clicking a script file from the explorer in Windows.

From the GUI, select the File menu, and select Open.



Select your file and it will display in the editor window. It is possible to select multiple files at this point. See the image below.

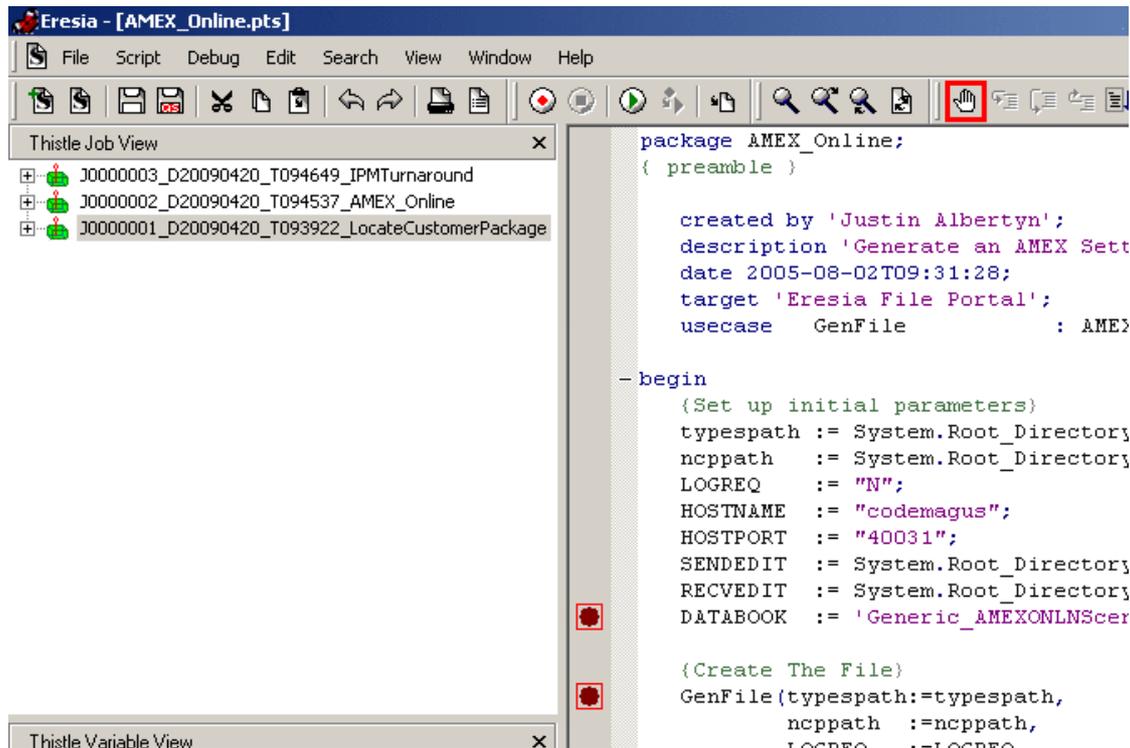


3.2 Setting breakpoints

Breakpoints are used to pause the execution of a script when it gets to a specific line of a script. This is true only when the user runs the script in Debug mode.

To set a breakpoint, select the line of the script you want to set the break on and then either press F9, or select the Debug/Toggle Breakpoint from the menu.

See the below image.



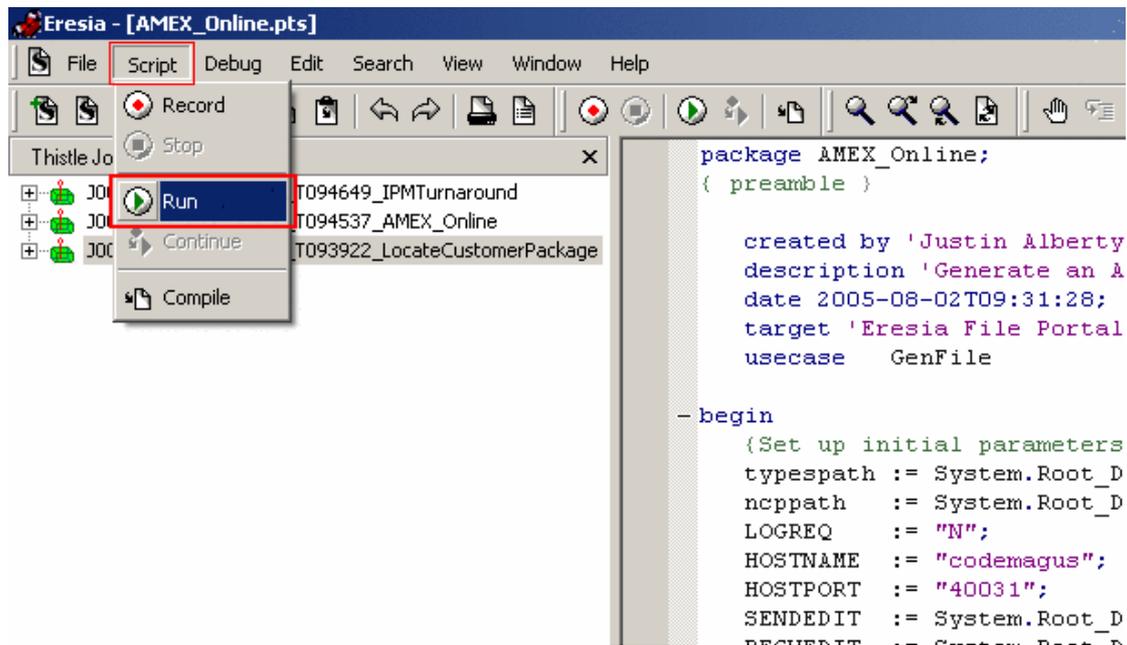
3.3 Running Eresia scripts

To run a script, the user can either run the script in debug mode, or run the script in non-debug mode. In debug mode, Eresia will stop whenever it reaches a point in a script where a break-point has been set. In non-debug mode, the breakpoints are ignored.

3.3.1 Running Eresia in non-debug mode

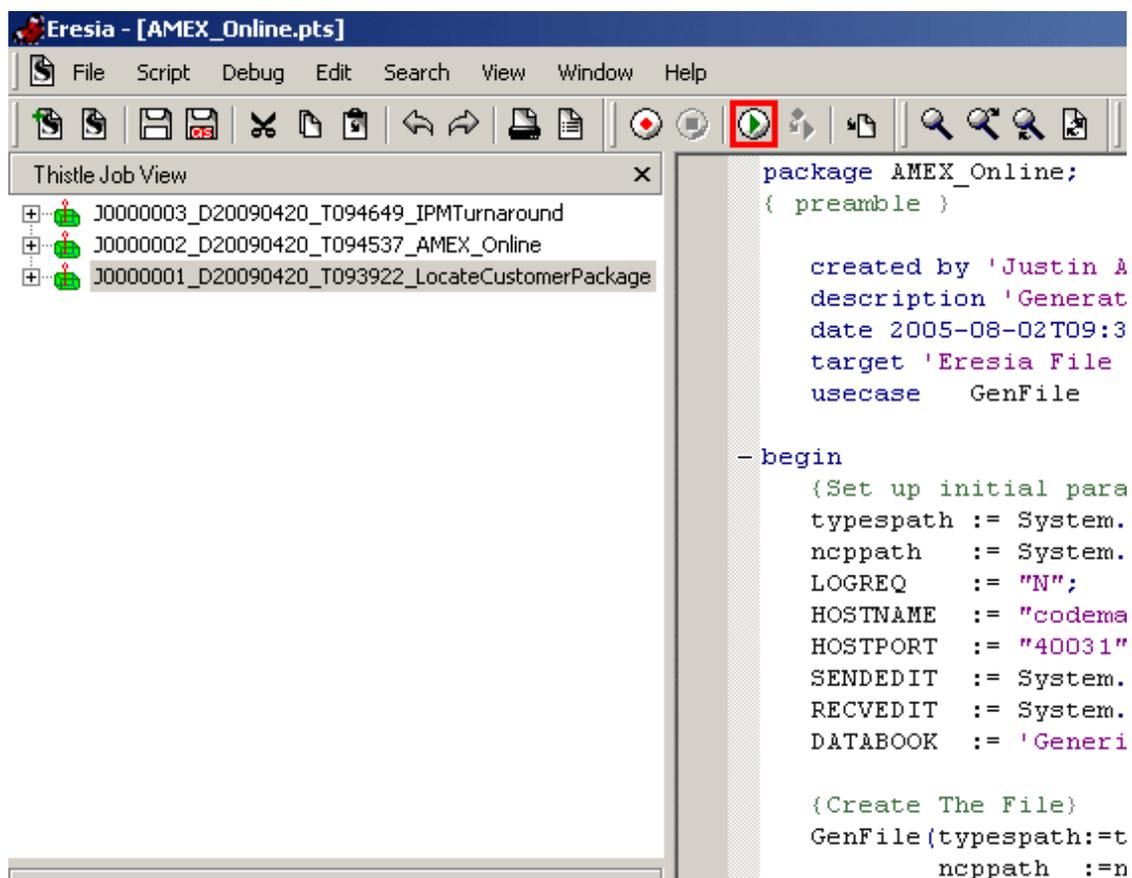
There are a number of ways to run a script in non-debug mode.

Select Script from the menu bar, and then select 'Run'. See the image below.



OR

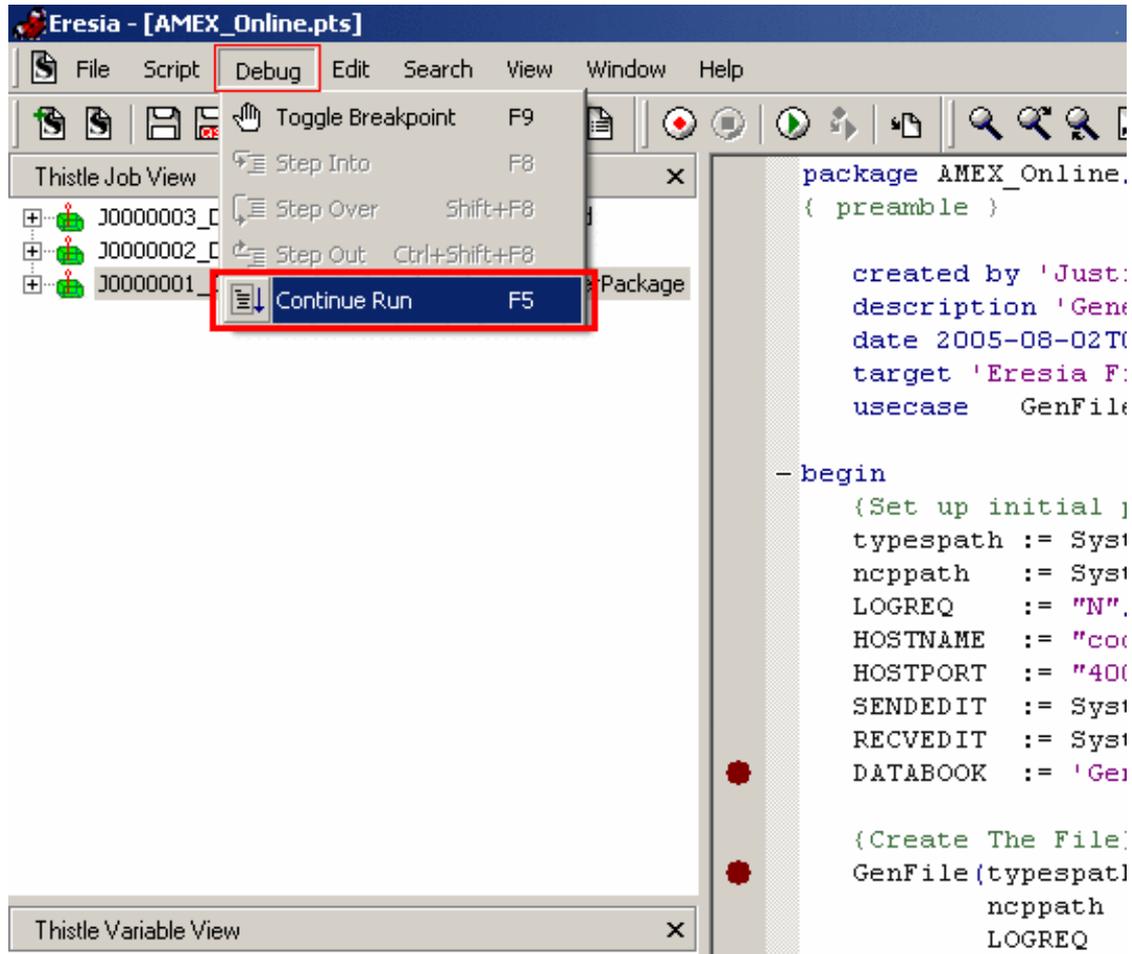
Press the 'play' button, as in the image below.



3.3.2 Running Eresia debug mode

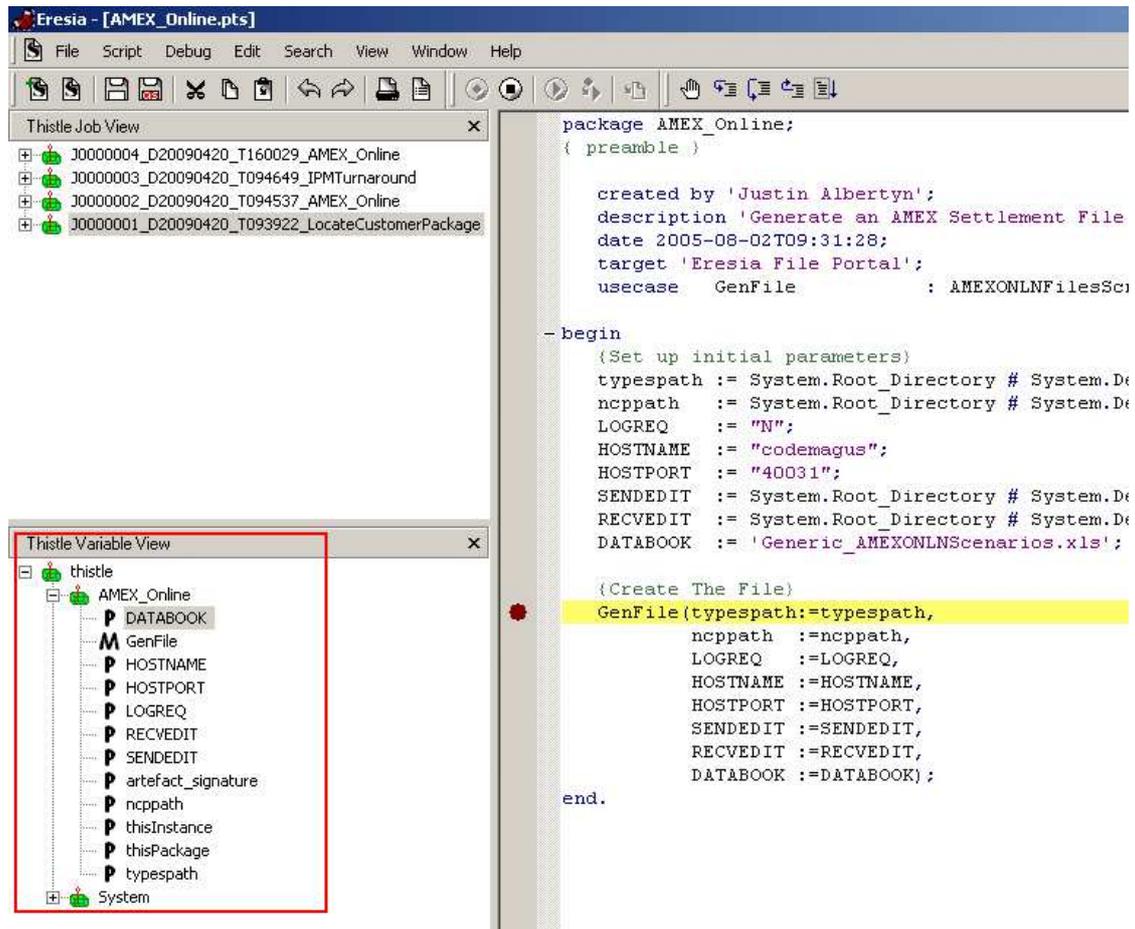
To run in debug mode, select Debug/Continue Run from the menu, or press F5.

See the image below.



3.4 Viewing Thistle variables during execution

When a breakpoint has been hit, the user can view the value of any thistle variable. This is done in the Variable View area (the red area in the below image).



Right-click the variable and select 'Value'.

The screenshot displays the Eresia IDE interface. At the top, the title bar reads "Eresia - [AMEX_Online.pts]". Below it is a menu bar with "File", "Script", "Debug", "Edit", "Search", "View", "Window", and "Help". A toolbar with various icons is positioned below the menu bar.

On the left side, there are two panels:

- Thistle Job View:** A tree view showing four job entries:
 - J0000004_D20090420_T160029_AMEX_Online
 - J0000003_D20090420_T094649_IPMTurnaround
 - J0000002_D20090420_T094537_AMEX_Online
 - J0000001_D20090420_T093922_LocateCustomerPackage
- Thistle Variable View:** A tree view showing the variable structure for the "thistle" package. Under "AMEX_Online", the "DATABOOK" variable is highlighted with a red box, and a "Value" button is visible next to it. Other variables listed include GenFile, HOSTNAME, HOSTPORT, LOGREQ, RECVEDIT, SENDEEDIT, artefact_signature, ncppath, thisInstance, thisPackage, and typespath.

The main editor area on the right contains the PL/SQL code for the "package AMEX_Online;". The code includes a preamble, a description, a date, a target, and a usecase. It defines several variables and a procedure "GenFile".

```

package AMEX_Online;
{ preamble }

created by 'Justin Albertyn';
description 'Generate an AMEX Settlement Fi
date 2005-08-02T09:31:28;
target 'Eresia File Portal';
usecase GenFile : AMEXONLNFiles

-begin
{Set up initial parameters}
typespath := System.Root_Directory # System
ncppath := System.Root_Directory # System
LOGREQ := "N";
HOSTNAME := "codemagus";
HOSTPORT := "40031";
SENDEEDIT := System.Root_Directory # System
RECVEDIT := System.Root_Directory # System
DATABOOK := 'Generic_AMEXONLNScenarios.xls

{Create The File}
GenFile(typespath:=typespath,
ncppath :=ncppath,
LOGREQ :=LOGREQ,
HOSTNAME :=HOSTNAME,
HOSTPORT :=HOSTPORT,
SENDEEDIT :=SENDEEDIT,
RECVEDIT :=RECVEDIT,
DATABOOK :=DATABOOK);

end.
    
```

4 Example Scripts

The following is an example of a Thistle Package script, and the associated Thistle Usecase script.

Below is the listing for SendAndReceiveMessage.uts.

```

usecase SendAndReceiveMessage();

{ preamble }

    created by 'Justin Albertyn';
    description 'Perform Amex Online Transactions';
    date 2005-08-04T10:51:18;
    target 'Eresia Network Injection Portal';
    interface Portal.NCP                : CodeMagus.NCP;
    interface Portal.Types               : CodeMagus.Types;
    interface Portal.Excel               : CodeMagus.Excel;

begin

    typespath := "C:\\CodeMagus\\CodeMagus\\EXAMPLEFormats"
                # "\\testdata\\objtypes\\example.objtypes";
    ncppath   := "C:\\CodeMagus\\CodeMagus\\bin\\thisncp.dll";

    ncp       := Portal.NCP.Connect(ncppath);
    types     := Portal.Types.Connect(typespath);

    {Connect to Excel}
    [thisInstance].TESTDATA := Portal.Excel.Connect(System.Root_Directory
                                                    # System.Defined_Names.FilesSpreadsheets #
                                                    DATABOOK);

    {Get All General Data From the Spreadsheet}
    HOSTNAME     := TESTDATA.WorkSheet.ExampleWorksheet.B[5];
    HOSTPORT    := TESTDATA.WorkSheet.ExampleWorksheet.B[6];
    IndexFrom   := TESTDATA.WorkSheet.ExampleWorksheet.B[7];
    IndexTo     := TESTDATA.WorkSheet.ExampleWorksheet.B[8];

    ncp.open(HOSTNAME :=HOSTNAME,
            HOSTPORT :=HOSTPORT);

    {Get necessary data from the spreadsheet as many times as necessary}
    for Index := IndexFrom to IndexTo do
    begin
        status := TESTDATA.WorkSheet.AMEX_ONLN_Sheet.A[Index];

    {Get All General Data From the Spreadsheet}

```

```
UserName      := TESTDATA.WorkSheet.ExampleWorksheet.D[Index];
Password      := TESTDATA.WorkSheet.ExampleWorksheet.E[Index];
AccountNumber := TESTDATA.WorkSheet.ExampleWorksheet.F[Index];

if (status = "EXECUTABLE") then
begin

    with MSG2SEND do begin
        UNAME    := UserName;
        PWORD    := Password;
        ACCNO    := AccountNumber;
        DESC     := "Send This Transaction Using The NIP.";
        VERSION  := 2;
        ACTION   := "Credit";
    end

    mymsg.buffer := types.GetBuffer(MSG2SEND, "MYTYPENAME");
    ncp.send(mymsg.buffer);
    actual.buffer := ncp.receive();
end
end
end.
```