

CA Gen

Diagram Trace Utility User Guide

Release 8.5



This Documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NON-INFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2013 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Gen
- AllFusion® Gen

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information that you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following resources:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Providing Feedback About Product Documentation

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

To provide feedback about CA Technologies product documentation, complete our short customer survey which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introducing the Diagram Trace Utility 9

Audience	9
Diagram Trace Utility Overview	10
Target Environments	11
Issues for Existing Applications Only	12
Issues for New and Existing Applications	12
Debugging Components	13
Modification of PAD View Data	13
Modification of System Variables	13
Expressions	14
Markers	15
Highlight PAD Code Fragments	16
Region Expansion and Contraction	16
National Language Support Capability	16
Accessibility Features	16
Where to Look for Additional Information	17

Chapter 2: Installing the Diagram Trace Utility 19

Hardware and Software Requirements	19
Installation Considerations	19
Diagram Trace Utility Installation	19
Post-installation Tasks	20

Chapter 3: Navigating the Diagram Trace Utility 21

Diagram Trace Utility Main Window	21
Default Perspective	22
Current Perspective	23
Reinstate the Default Perspective	25
Main Menu Bar	25
Main Window Toolbar	29
Views	29
Activate and Close a View	30
Rearrange Views	30
Rearrange Tabbed Views	32
Rearrange a Group of Tabbed Views	33
Open a View	35

The Bookmarks View	37
The Breakpoints View.....	39
The Console View	43
The Debug View	44
The Diagram Trace Explorer View.....	50
The Error Log View	51
The Expressions View	53
The Outline View	60
The Variables View.....	62
PAD Code View Editors.....	67
The Vertical Ruler and Pop-up Menu	68
Activate a PAD Code View Editor	69
Add a New PAD Code View Editor.....	69
Tile PAD Code View Editors	70
Rearrange the Order of Tabbed PAD Code View Editors	71
Import Preferences or PAD Files into a Debug Session.....	72
Export Preferences to a File	75
Use Online Help	77
Context-Sensitive Help	78
F1 Support	78
Dynamic Help.....	78

Chapter 4: Preparing to Debug 83

Configure the Generated Application.....	83
Start the Diagram Trace Utility	84
Exit the Diagram Trace Utility	84
Start a Debug Session.....	85
Stop a Debug Session.....	85
Configure the Diagram Trace Utility	85

Chapter 5: Running a Debug Session 87

Debug Session Overview.....	88
Configure and Start a Debug Session.....	89
Start the Generated Application	89
Prerequisites	90
Prepare the Application for Trace.....	90
Start the Generated Application on Windows	91
Start the Generated Application on UNIX and Linux	92
Start the Generated Application on NonStop	93
Verify the Diagram Trace Utility Connection.....	93
How a Traced Program Executes	94

Use Breakpoints to Control Program Execution	94
Add a Line Breakpoint	96
Enable a Breakpoint.....	96
Disable a Breakpoint.....	96
Remove a Breakpoint in the Breakpoints View	97
Remove a Breakpoint from the Vertical Ruler of a PAD Code View Editor	97
Set the Suspend on Initial Entry Preference	98
Set Hit Counts and Conditional Breakpoints	98
Manage Bookmarks	104
Add the Bookmarks View to the Current Perspective	104
Add a Bookmark	105
Access an Action Diagram with a Bookmark	105
Remove a Bookmark.....	106
Exit the Debug Session	106

Chapter 6: Customizing the Diagram Trace Utility 107

Change the Diagram Trace Utility Main Window Layout.....	107
Set Diagram Trace Preferences.....	107
Set General Startup Preferences for Diagram Trace Utility.....	108
Set Diagram Trace PAD Code View Editor Preferences	110
Set Diagram Trace Editor Folding Preferences	112
Set Editor Syntax Coloring Preferences	114
Set Diagram Trace Step Filtering Preferences	115
Set General Preferences	117
Set General Appearance Preferences for Editor and View Tabs	119
Set General Appearance Preferences for Colors and Fonts	122
Set General Preferences for PAD Code View Editors.....	124
Set General Preferences for Text Editors.....	126
Set Accessibility Preferences for Text Editors.....	128
Set Annotations Preferences for Text Editors	129
Set General Preferences for Keys.....	131
Export Key Preferences to a File.....	132
Set Help Preferences	134
Set Run/Debug Preferences	136

Chapter 7: Troubleshooting 139

Console and Error Logs.....	139
Review Contents of the Console Log View	139
Review Contents of the Error Log View	141
View Error Details.....	142
View the Detail Log for a Selected Error in Notepad	143

Import an Error Log.....	144
Export an Error Log	144
Clear and Restore an Error Log	144
Troubleshoot Connection Problems	145
Troubleshoot Configuration Problems	146

Appendix A: Using Accessibility Keys **147**

Access the Menus.....	147
Access the Controls.....	148
Switch Between PAD Code View Editors and Views	148
Set the Color Preferences	148
Navigate the Online Help System	149
Change the Help Browser Settings	150

Index **153**

Chapter 1: Introducing the Diagram Trace Utility

This section contains the following topics:

[Audience](#) (see page 9)

[Diagram Trace Utility Overview](#) (see page 10)

[Target Environments](#) (see page 11)

[Issues for Existing Applications Only](#) (see page 12)

[Issues for New and Existing Applications](#) (see page 12)

[Debugging Components](#) (see page 13)

[Where to Look for Additional Information](#) (see page 17)

Audience

The *Diagram Trace Utility User Guide* is intended for CA Gen application developers who need to debug or analyze code flow through the CA Gen Procedure Action Diagram (PAD) language statements in client, server, or stand-alone block mode applications.

This guide contains information on installing, configuring, and using the Diagram Trace Utility.

Important! Only the Diagram Trace Utility features that CA supports are documented in this guide. You may see more features in menus, views, or toolbars than CA supports since the utility is implemented as a Rich Client Platform (RCP) based on the Eclipse Foundation Framework. Using those unsupported features may produce unpredictable results. For more information on the Eclipse Foundation Framework, go to this URL: <http://www.eclipse.org/>

This guide assumes that you have a good working knowledge of the CA Gen product and know how to perform the following tasks:

- Develop CA Gen applications
- Use the CA Gen Toolset or Encyclopedia to generate application code
- Use the CA Gen Procedure Action Diagram programming language

Diagram Trace Utility Overview

The Diagram Trace Utility is an interactive debugging tool for applications written in C, Java, or C# languages that are generated with the CA Gen code generator and run on Microsoft Windows, UNIX, or Linux operating systems. You cannot use the Diagram Trace Utility to debug handwritten CA Gen applications or applications that are generated with code generators other than the CA Gen Toolset and Encyclopedia code generators.

CA Gen applications are model-driven applications that are developed using CA Gen Procedure Action Diagram (PAD) language statements. The Diagram Trace Utility operates at the PAD statement level, which is the CA Gen natural programming language level, instead of the C, Java, or C# language level. The generated application or portion of the application being traced must have been generated with Tracing enabled.

Note: If you require source-level debugging at the C, Java, or C# language level, you must use a third-party, native language debugger.

During code generation, the PAD language statements are converted to appropriate programming language statements based on the selected target language (C, Java, or C#). Using the Diagram Trace graphical user interface (GUI), you can trace or step through the execution of statements in the PAD and examine the contents of views and system attributes to locate the problems in your application. The utility lists the Action Diagram statements contained in the Action Diagram where the controlled application is suspended.

Communication between the application being traced and the Diagram Trace Utility server occurs through the exchange of a series of command and control messages. These messages pass variable data, commands, status, and statement source code strings between the application and the utility. For each PAD statement that is executed, the following actions occur:

1. The executing application calls the CA Gen runtime trace function.
2. The trace function sends the following data to the Diagram Trace Utility server for each statement contained in the application:
 - The statement location
 - The current values of all CA Gen system data
 - The current values of all active Import, Export, Entity, and Local views
3. If needed, the Diagram Trace Utility requests detailed information from the application runtime. The detailed data can include:
 - Action Diagram PAD text
 - View descriptions
 - View data

4. The Diagram Trace Utility returns control to the application runtime and indicates whether to execute or skip the next statement.
5. The application runtime returns control to the executing application and indicates whether to execute or skip the next statement.
6. Application execution ends or Steps 1-5 are repeated.

The following information on the communications to and from the Diagram Trace Utility is logged on the application side:

- Connection attempts (success or failure)
- Each Action Block statement encountered
- Variable change value request from the Diagram Trace Utility
- Disconnect and termination events

When the Diagram Trace Utility and the application being traced reside on separate machines, network latency may cause long delays and give the appearance that the Diagram Trace Utility is locked up. In this case, we recommend that you generate only the action blocks that you want to debug with Trace selected. This will reduce the traffic between the application and the Diagram Trace Utility, thus minimizing this latency effect.

The Diagram Trace Utility caches the PAD code data it receives from the application in its workspace subdirectory in the Diagram Trace Utility installation area. The utility caches the Action Diagram code based on the generation time shown on the first line of the Action Diagram.

Note: The application does not resend the source statements to the utility until the generation time in the application is different than the generation time in the cached version of the Diagram Trace Utility. Keep this in mind if you modify the generated source code.

Target Environments

The Diagram Trace Utility runs only on Microsoft Windows operating systems. You cannot run the Diagram Trace Utility on UNIX or Linux operating systems, although you can debug generated applications that run on UNIX or Linux operating systems. Detailed information on the specific Windows operating systems on which the Diagram Trace Utility is supported is documented in the *Technical Requirements*. You can obtain this document from the CA website: <http://ca.com/support>.

The Diagram Trace Utility supports application tracing using the following languages and client/server combinations:

- C language applications
 - C Servers
 - Transaction enabler (TE)
 - Tuxedo
 - WebSphere MQ
 - NonStop RSC/MP
 - GUI clients
 - CA Gen Testing Tool
 - Block mode clients
- Java Language applications
 - Web Generation clients
 - Enterprise Java Bean (EJB) servers
- C# language applications
 - .Net Servers
 - ASP.NET Web Clients

Issues for Existing Applications Only

The Diagram Trace Utility does not support tracing applications that have been generated with CA Gen code generators in releases prior to release 7.6. To be able to use the Diagram Trace Utility, you must regenerate the existing application with AllFusion 7.6 code generators or later.

Issues for New and Existing Applications

When GUI client applications are traced with the Diagram Trace Utility, a difference may be noted with the client application control that has the current keyboard focus. The difference is whether or not the control with focus is highlighted.

To enable trace functionality the current keyboard focus moves from the GUI client application to the Diagram Trace Utility and back as the application is traced. To avoid focus problems when interacting with the Diagram Trace Utility, the GUI client application's control highlighting may be lost. This is expected behavior that should not be seen when the GUI client application is not being traced.

Debugging Components

The Diagram Trace Utility contains the following features to help you debug your applications:

- [Modification of PAD view data](#) (see page 13)
- [Modification of system variables](#) (see page 13)
- [Evaluating defined expressions](#) (see page 14)
- [Setting breakpoint and bookmark markers](#) (see page 15)
- [Highlighting PAD Code fragments](#) (see page 16)
- [Region expansion and contraction](#) (see page 16)
- [National Language Support capability](#) (see page 16)
- [Accessibility features](#) (see page 16)

Modification of PAD View Data

The Diagram Trace Utility lets you access and modify the PAD import, export, entity, and local view data. You can inspect data values and change them if required. The utility validates any changed data to ensure the type and length of data entered adheres to the variable's defined attribute type.

Note: The Diagram Trace Utility does not validate application-defined edit patterns during view data modification. The utility also does not validate any changed data for permitted value violation.

Note: You cannot modify BLOB data using the Diagram Trace Utility.

Modification of System Variables

During a Diagram Trace debugging session, you can access and modify the following CA Gen system variables:

- Command
- ExitStateName
- ExitState
- TranCode
- ExitStateInfoMsg
- UserID
- TerminalID

- PrinterID
- Dialect
- ClientUserID
- NextLocation
- CurrentDate
- CurrentTime
- ClientPassword
- NextTran
- TransactionRetryLimit
- TransactionRetryCount

Note: You can view the ExitStateName system variable, but you cannot modify it.

The ExitStateName and ExitStateInfoMsg are not supported for Web View applications.

You can inspect the data values and change them if required. The Diagram Trace Utility validates changed data to ensure the type and length of data entered adheres to attribute types defined for variables.

Expressions

An expression is a snippet of code that you can evaluate to produce a value. You may need to evaluate some expressions at a specific location in the program so that the Diagram Trace Utility can reference their values.

Expressions can be as complex as needed. Essentially, complex expressions can have unlimited lengths. The complexity of the expression is limited only by the availability of memory in the underlying Java virtual machine (JVM).

Watch expressions are expressions that are repeatedly evaluated as the program executes. You can use Watch expressions to implement Watch lists that show changes in the values of expressions as the program executes.

In expressions, operator precedence follows the standard arithmetic precedence. An expression can contain one or more combinations of the following operators:

- Boolean: True/False
- Simple mathematical operators: =, -, /, *
- Comparison operators: <, !<, <=, !<=, >, !>, >=, !=, ==
- Logical operators: ||, &&

- Expressions contained in parentheses:()
- Negation Operator: "-"

String literals must be contained within double quotes ("""). Text strings that are not contained within double quotes are assumed to be variable types instead of literal types. Arguments within double quotes are never evaluated.

The following list contains examples of different combinations of operators:

- `Var1 == True`
`Var2 != False`
- `Var3 + Var4 < Var5 + Var4`
- `Var2 <= 33` (constant value of 33)
- `Var3 <= MyVal` (variable MyVal)
- `Var 3 >= "MyVal"` (constant string "MyVal")
- `Var1 + Var2 * 3.21 == 9.664`
- `((Var2 + 6) + 2) / 99`
- `(Var1 + 1) <= Var2`
- `(Var1 + (Var2 * 3) != 999`

Markers

Markers are objects that you can associate with specific lines of Diagram Trace PAD code. You can create the following types of markers during a debug session:

Bookmarks

Used to place anchors on a specific line of PAD code in a debug session. You can use these anchors as quick links to move from one source position or file to another. You can view all bookmarks that you have created in the Bookmarks view.

Breakpoints

Used to place markers on specific PAD statement lines to suspend program execution whenever the session encounters a breakpoint marker. You can view all breakpoints that you have created in the Breakpoints view.

You can set breakpoints at the Action Diagram level and at the Window GUI Event method level to stop:

- At a specific PAD statement line
- At a specific PAD statement line if a user-defined condition is satisfied
- At the entrance of or just before the exit of an Action Diagram
- When a specified data item changes value

Highlight PAD Code Fragments

You can define your own color scheme to highlight PAD code fragments so you can locate them easily during your debugging session. Use highlighting to help you see the following PAD code fragments:

- Keywords
- View types (import, export, entity, and local)

Use the PAD Code View editor to display highlighted PAD code fragments.

More information:

[PAD Code View Editors](#) (see page 67)

Region Expansion and Contraction

You can expand or contract code view regions dynamically as required. This capability lets you control the amount of data displayed in the PAD Code View window at any given time and reduce clutter caused by code segments that are of no interest.

National Language Support Capability

The Diagram Trace Utility supports any NLS data that can be specified with the CA Gen PAD language statements.

Accessibility Features

Accessibility features help people with physical disabilities, such as restricted mobility or limited vision, to use software products successfully.

The Diagram Trace Utility provides the following accessibility features:

- You can operate all features of the Diagram Trace Utility using the keyboard instead of a mouse.
- You can set all fonts and colors in the Diagram Trace Utility using the Colors and Fonts Preferences dialog.

Note: The Diagram Trace Utility does not support screen reader software or the use of Microsoft Active Accessibility (MSAA) APIs to render user interface elements accessible to assistive technology.

More information:

[Using Accessibility Keys](#) (see page 147)

Where to Look for Additional Information

The Diagram Trace Utility provides online help during your debugging sessions. You can activate the help from the Help menu.

For information on executing CA Gen application programs with Diagram Trace Utility, see the following CA Gen guides:

- *ASP.NET User Guide*
- *Block Mode Design Guide*
- *Build Tool User Guide*
- *Client Server Design Guide*
- *Distributed Processing - .NET Server User Guide*
- *Distributed Processing - Enterprise JavaBean User Guide*
- *Distributed Processing - Overview Guide*
- *Distributed Processing - WebSphere MQ User Guide*
- *UNIX and Linux Implementation Toolset User Guide*
- *Windows Implementation Toolset User Guide*
- *NonStop Implementation Toolset User Guide*
- *Rapid Test Tool Help*
- *Transaction Enabler User Guide*
- *Tutorial*
- *Tuxedo User Guide*
- *Web Generation User Guide*

Chapter 2: Installing the Diagram Trace Utility

Before you install the Diagram Trace Utility, ensure you review information on hardware and software requirements, considerations for installing the Diagram Trace Utility, and post-installation tasks.

This section contains the following topics:

[Hardware and Software Requirements](#) (see page 19)

[Installation Considerations](#) (see page 19)

[Diagram Trace Utility Installation](#) (see page 19)

[Post-installation Tasks](#) (see page 20)

Hardware and Software Requirements

The hardware and software requirements are documented in the *Technical Requirements*. You can obtain this guide from the CA website: <http://ca.com/support>

Installation Considerations

Before you install the Diagram Trace Utility, review the following considerations:

- The Diagram Trace Utility is implemented on the Java-based Eclipse Workbench and is written in the Java programming language. For more information on the Eclipse Foundation Framework, go to this URL: <http://www.eclipse.org/>
- Although the Diagram Trace Utility is built on the Eclipse Workbench foundation, no additional Eclipse installation package is required.
- The underlying Eclipse Workbench provides support for execution on platforms other than Windows platforms. However, the Diagram Trace Utility runs only on Microsoft Windows operating systems.

Diagram Trace Utility Installation

The Diagram Trace Utility is a separately installable feature in the CA Gen feature set. Both the Typical and Custom installation options install the Diagram Trace Utility. This utility has no dependency on, nor is it a dependent of, any other CA Gen product.

Important! Diagram Trace maintains a set of support files in the Diagram Trace installation directory and subdirectories that represent your workspace. Modifying these support files outside the Diagram Trace Utility may produce unintended side effects or loss of Diagram Trace functionality.

For more information about installing the utility and product licensing, see the following CA Gen guides:

- *Distributed Systems Installation Guide*

Post-installation Tasks

You can start the Diagram Trace Utility by selecting Start, All Programs, CA, Gen <version>, Diagram Trace Utility. You can also start the Diagram Trace Utility by using %GENxx%Gen\DiagramTrace.bat. In either case, the Diagram Trace Utility uses the Java Runtime Environment referenced by %GENxxJRE%.

Note: xx denotes the current CA Gen release number. For the current release number, see *Release Notes*.

You can customize a number of preferences; however, customization is not required for successful Diagram Trace Utility operation.

More information:

[Configure the Diagram Trace Utility](#) (see page 85)

Chapter 3: Navigating the Diagram Trace Utility

Before you begin a debug session, it is important to understand how to navigate the Diagram Trace Utility main window and perform basic actions.

This section contains the following topics:

[Diagram Trace Utility Main Window](#) (see page 21)

[Views](#) (see page 29)

[PAD Code View Editors](#) (see page 67)

[Import Preferences or PAD Files into a Debug Session](#) (see page 72)

[Export Preferences to a File](#) (see page 75)

[Use Online Help](#) (see page 77)

Diagram Trace Utility Main Window

The Diagram Trace Utility contains a group of the following items in the main window that provide all the functions you need to debug your applications:

- Views
- PAD Code View Editors
- Menus
- Toolbars

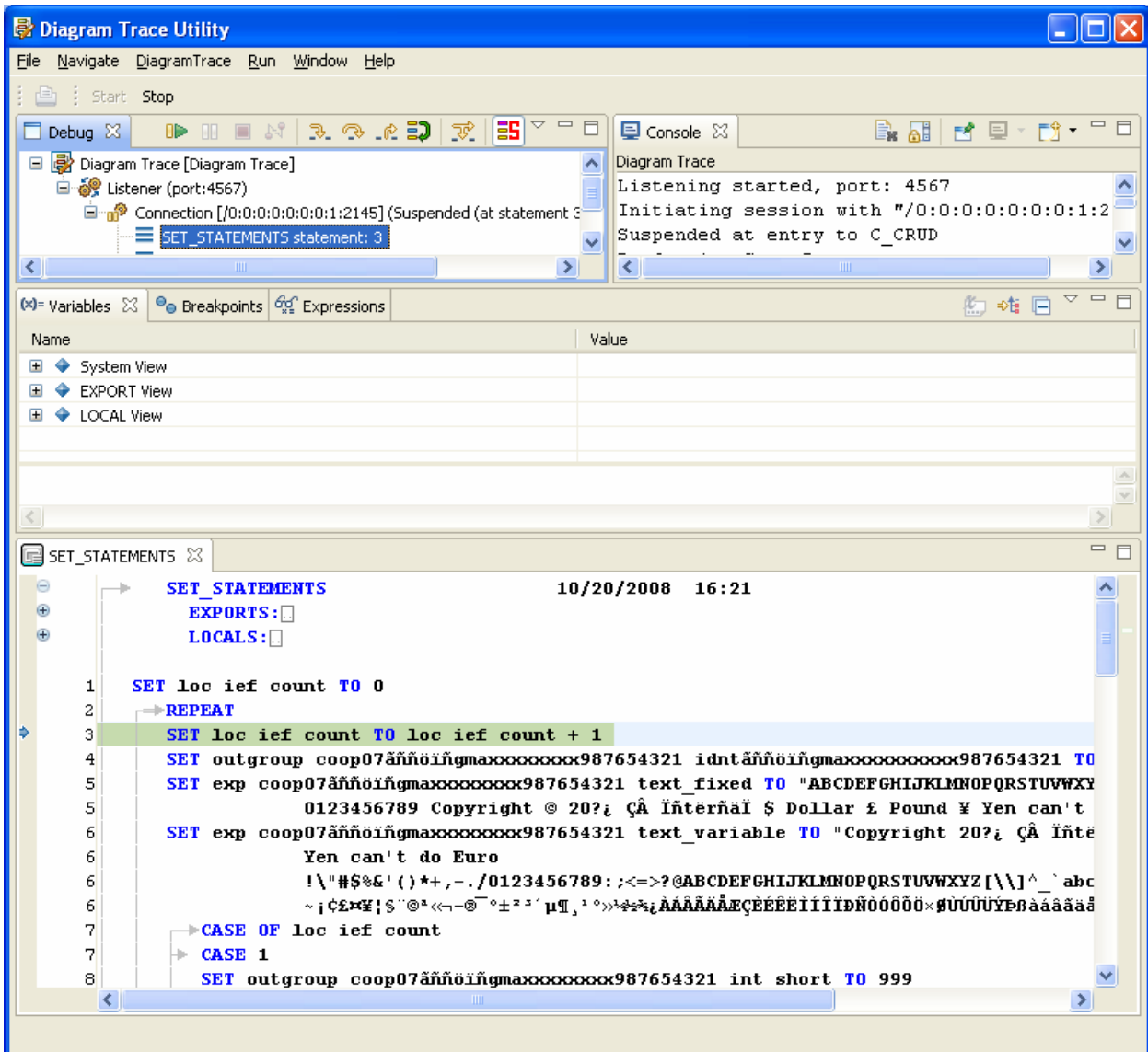
You can change the default window layout, or perspective, by opening views, closing views, or docking views in different positions in the main window.

More information:

[Configure the Diagram Trace Utility](#) (see page 85)

Default Perspective

The following illustration is an example of a default perspective (window layout).



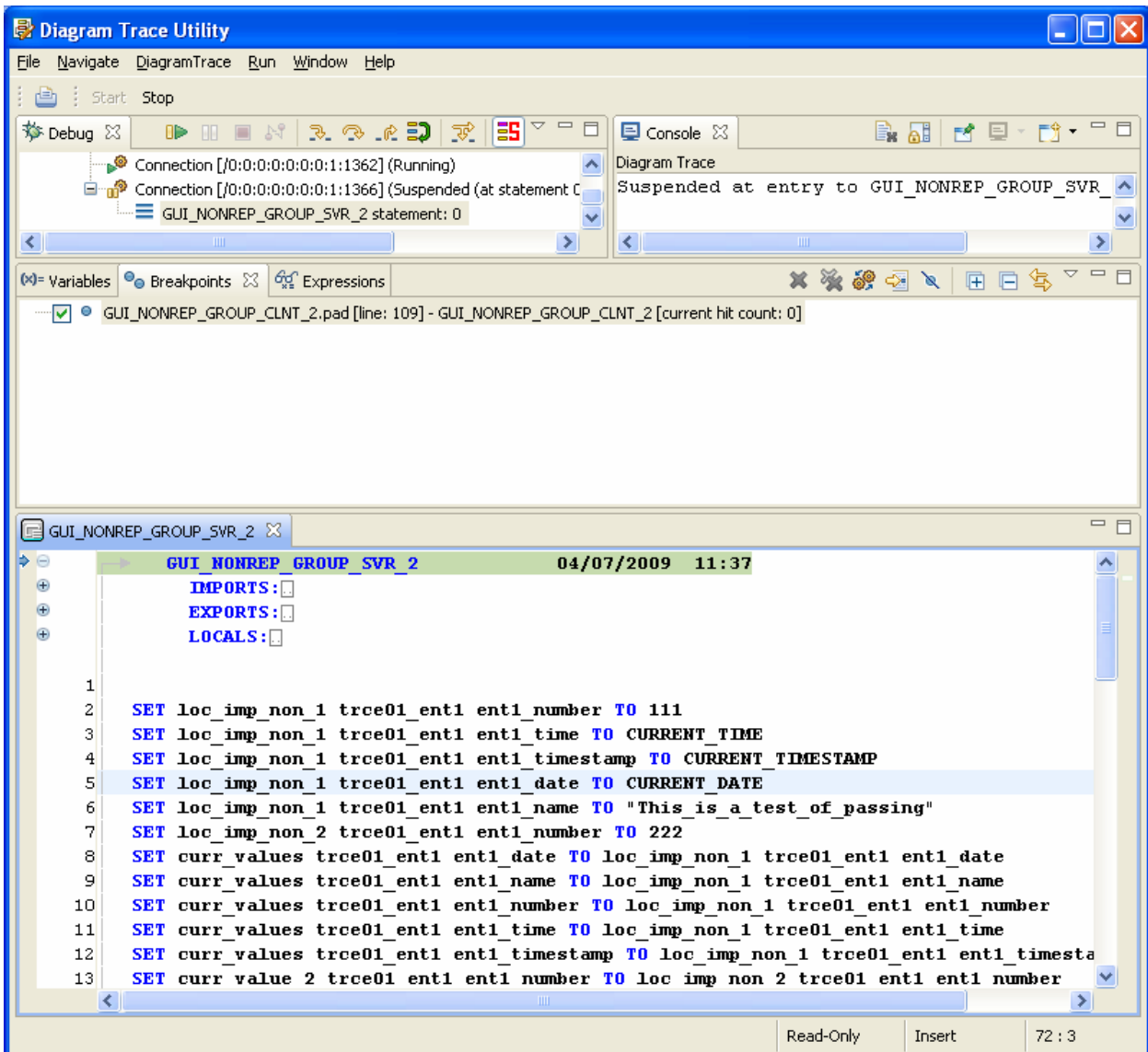
The Diagram Trace Utility is preconfigured to start a debugging session automatically when you start the software. The first time you start the Diagram Trace Utility, it displays the default perspective, which contains the Debug view, Console view, Variables view, Breakpoints view, Expressions view, and the PAD Code View editor. You can change the default window layout to suit your needs.

Note: The first time you start the Diagram Trace Utility, the views contain no data, except for the Debug view and the Console view. The Debug view contains a Diagram Trace Listener process that listens on the default port for new application connections. The Console view contains information about the session initiation and shows the port on which the Listener process is listening for connections.

Current Perspective

When you change the default layout of the main window, the Diagram Trace Utility automatically saves the new layout configuration as the *current* perspective when you exit the window. Each time you start the Diagram Trace Utility, the current perspective is displayed, unless you restore the default perspective. You can change the window layout by opening new views, closing views, and rearranging views.

The following illustration shows a current perspective that you can create by closing several views and rearranging the remaining views. In this example, the PAD Code View editor is highlighted, which indicates that it is currently the active component in the main window of the Diagram Trace Utility.



Note: View and PAD Code View editor window components can be active or inactive, but only one component can be active at any one time.

Reinstate the Default Perspective

When you reset the default window perspective, the current perspective you created previously is no longer available for viewing.

Follow these steps:

1. Select Reset Perspective from the Window menu on the Diagram Trace Utility main window.

The Reset Perspective dialog appears.

2. Click OK.

The current Diagram Trace Utility perspective is reset to its default settings.

Main Menu Bar

The Diagram Trace Utility main window contains a main menu bar on the upper-left side of the window that contains the File, Navigate, DiagramTrace, Run, Window, and Help menus.

File Menu

You can use the commands on the File menu to close or print PAD Code editor views, import or export preferences, or close the Diagram Trace Utility.

Close

Closes the currently or last selected PAD Code View editor.

Close All

Closes all currently opened PAD Code View editors.

Print

Prints the content of the currently selected PAD Code View editor.

Import

Imports a generated CA Gen Action Diagram File or imports Diagram Trace preference settings from a previously created preference export file.

Export

Exports the set of Diagram Trace preferences to a file.

Exit

Exits the Diagram Trace Utility.

Navigate Menu

You can use the Go to Line command on the Navigate menu to move your cursor to a specified line of code in the PAD Code View editor.

Go to Line

Displays the Go to Line dialog that lets you specify a line number to move the cursor to in the PAD Code View editor. The upper limit of line numbers that you can specify depends on the size of the Action Diagram displayed in the PAD Code View editor.

Note: This option is available only when the PAD Code View editor is the active view.

Diagram Trace Menu

You can use the commands on the DiagramTrace menu to start or stop communication to the traced application.

Start

Starts Diagram Trace listening on the configured port for communication from an application to be traced.

Stop

Terminates the connection to the traced application.

Run Menu

You can use the commands on the Run menu to enable, disable, skip, or remove breakpoints. You can also resume, suspend, terminate, watch, or inspect the executing application.

Toggle Line Breakpoint

Toggles the current state of the Line Breakpoint marker. Sets a breakpoint if not currently set and clears a Breakpoint if currently set.

Toggle Method Breakpoint

Toggles the current state of the Method Breakpoint marker. Sets a breakpoint if not currently set, clears a breakpoint if currently set.

Toggle Watchpoint

Toggles the current state of the watchpoint. Creates a watchpoint if not currently set or clears an existing watchpoint.

Skip All Breakpoints

Causes Diagram Trace to ignore the current setting of all breakpoints allowing the application to run without breaking.

Remove All Breakpoints

Deletes all breakpoints.

Resume

Resumes execution from the statement on which the application execution stopped.

Suspend

Suspends execution on one or more selected connections.

Terminate

Terminates the selected debug target and ends the debug session.

Step Into

Resumes execution, stopping at the next statement either in the same method, a called method, or (if you are at the end of a method) in the method from which the current method was called.

Step Over

Resumes execution, stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Step Return

Resumes execution, stopping at the next statement in the method from which the current method was called.

Skip Over

Resumes the application (without executing the current statement), stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Skip To Line

Skips the execution of a sequence of statements from the current line of code to the line containing the cursor.

Run To Line

Resumes execution of a suspended program until the execution reaches the specified line. By doing this, you can suspend execution at a line without setting a breakpoint. Program execution resumes and suspends just before the specified line executes.

Use Step Filters

Toggles the Use Step Filters button in the Debug view toolbar. When the action is toggled on, each of the step actions (over, into, return) applies the set of step filters defined through the Window menu Preferences option.

Watch

Sets a Watch expression on selected text in the PAD Code View editor and displays the Watch expression in the Expressions view.

Inspect

Displays a pop-up Inspection window in the PAD Code View editor for the selected view reference and lets you move the selection to the Expressions view. This option is only available when the PAD Code View editor has the focus and has a matching suspended Stack Frame, and the text selection encompasses an entire view reference.

Window Menu

You can use the commands on the Window menu to show a particular view, reset the perspective, navigate in the displayed PAD Code View editors and views, and set the Preferences for the Diagram Trace Utility.

Show View

Displays a submenu listing the available views and you can click the appropriate view to open.

Reset Perspective

Changes the current perspective (or window layout) to the default built-in perspective.

Navigation

Displays a submenu for navigating in the displayed PAD Code View editors and views.

Preferences

Displays a dialog containing subpanels allowing access to various preferences used to configure the appearance of views and the behavior of the Diagram Trace Utility tools.

Help Menu

You can use the commands on the Help menu to access online help, CA online technical support and product version information.

Help Contents

Displays the help table of contents in a help window or external browser.

Search

Displays the help window opened on the Search page.

Dynamic Help

Displays the help window opened to the Related Topics page.

Online Support




Launches the default web browser and connects to the CA technical support website.

About Diagram Trace

Displays product information, links to CA home and technical support websites, and access to the End-user License Agreement.

Main Window Toolbar

The Diagram Trace Utility main window contains a toolbar bar below the main menu bar. This toolbar contains icons that you can click to perform the actions described in the following table:

Icon	Description
	Opens the Print dialog and prints the contents of the PAD being edited. This option is only available if a PAD Code view editor is open and the editor has the focus.
	<p>Starts a Diagram Trace Server and updates the Debug view with the new server. The Diagram Trace Utility starts a listener process for every port specified in the Listen On Port(s) field on the Diagram Trace Preferences dialog and removes information for the previously terminated Diagram Trace server.</p> <p>This option is only available when there is no Diagram Trace server started.</p>
	<p>Terminates the Diagram Trace server and removes the items from the list in the Debug view.</p> <p>This option is only available when a Diagram Trace Server is running.</p>

Views

Views are visual components of the Diagram Trace Utility that support PAD Code View editors and provide alternative presentations and ways to navigate the information being traced in your application. A view can appear by itself, or you can stack it with other views in a tabbed notebook in the Diagram Trace Utility main window. Views also have their own toolbars and pop-up menus that contain actions you can perform that are specific to the view.

The Diagram Trace Utility supports the following types of views:

- [Bookmarks](#) (see page 37)
- [Breakpoints](#) (see page 39)

- [Console](#) (see page 43)
- [Debug](#) (see page 44)
- [Diagram Trace Explorer](#) (see page 50)
- [Error Log](#) (see page 51)
- [Expressions](#) (see page 53)
- [Outline](#) (see page 60)
- [Variables](#) (see page 62)

You can use these views to navigate a hierarchy of information (such as variable values), open a PAD Code View editor, show breakpoints, bookmarks, expression watchpoints, console output, and various other activities. The Diagram Trace Utility immediately saves any changes you make in a view. Only one instance of a particular type of view can exist in a Diagram Trace Utility window at any given time.

Activate and Close a View

A view displayed in the Diagram Trace Utility main window can be active or inactive, but only one view can be active at a given time.

To activate a view, click the tab of the view you want to activate. The system highlights the selected view's title bar. The highlighted view becomes the active view.

Note: All views have Maximize and Minimize view buttons on their toolbars that you can click to expand and shrink the view.

To close a view, click the X in the upper right corner of the view tab.

Important! When you close a view tab, the view remains closed in the current debug session and future debug sessions unless you manually reopen the view.

Rearrange Views

You can rearrange the placement of views and editors in the Diagram Trace Utility main window anytime by dragging and dropping the view to dock (or place) it in a different location.

Follow these steps:

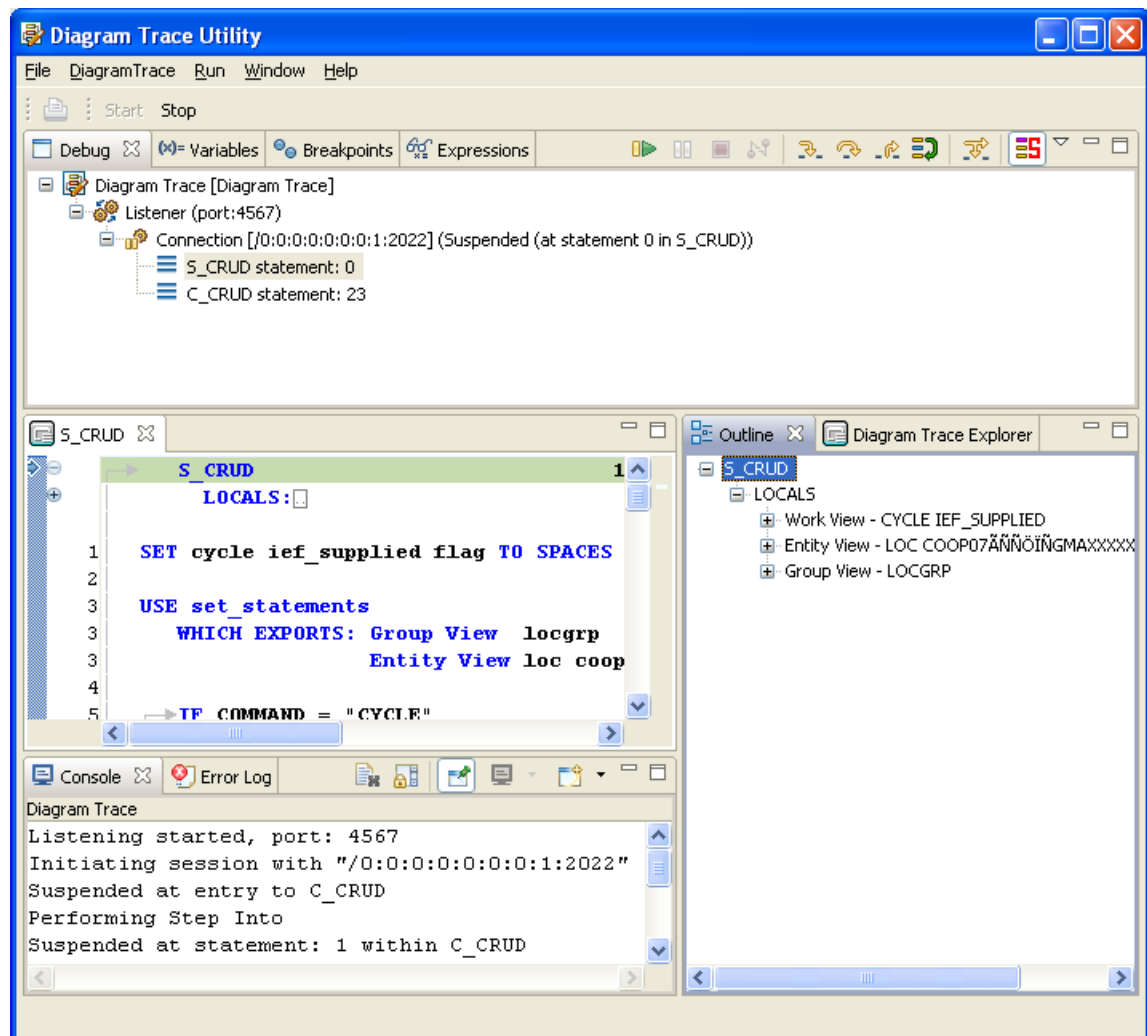
1. Click the title bar of the view you want to rearrange and drag the view across the main window.

Various [drop cursors](#) (see page 32) indicate where the view will dock in relation to the view or PAD Code View editor area beneath the cursor when you release the mouse button. In addition, a tool tip appears that provides additional feedback on the location in which the view will dock.

2. Release the mouse button to dock the view in any position in the main window. Drag the view's title bar to dock the view in another position in the main window.

The views are docked to the new positions in the main window.

Example: The Diagram Trace Explorer view is stacked with the Outline view into a tabbed notebook and the Outline view is the active view.



Types of Drop Cursors

As you drag a view across the Diagram Trace Utility main window, the drop cursors in the following list may be displayed to indicate where the view will be docked after you release the mouse button:

Up Arrow

Docks above the view that appears beneath the cursor.

Down Arrow

Docks below the view that appears beneath the cursor.

Right Arrow

Docks to the right of the view that appears beneath the cursor.

Left Arrow

Docks to the left of the view that appears beneath the cursor.

Stacked Cursor

Docks the view in the same pane as the view that appears beneath the cursor. The view displays as a tab in the pane.

Restricted Cursor (line through circle)

Does not dock the view at the location if you release the mouse there.

Example: You cannot dock a view in the PAD Code View editor area.

Detached Cursor (box with four quadrants)

Detaches the view from its location in the main window. You can re-attach the view by clicking the tab of the detached view and dragging and dropping the view at its original location or at another location on the main window.

Rearrange Tabbed Views

The Diagram Trace Utility main window lets you rearrange the order of views in a tabbed notebook. The following procedure explains how to rearrange the Variables and Breakpoints views in a tabbed notebook.

Follow these steps:

1. Select Reset, Perspective from the Window menu.

The window perspective is reset to its original layout.

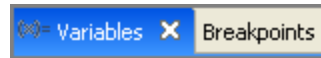
2. Drag the Variables title bar to the right of the Breakpoints tab until the cursor appears as a Stack cursor, then release the mouse button.

The Variables view tab now appears to the right of the Breakpoints tab.



3. Drag the Breakpoints view tab to the right of the Variables tab until the cursor is a Stack cursor and release the mouse button.

The Breakpoints view tab now appears to the right of the Variables tab.



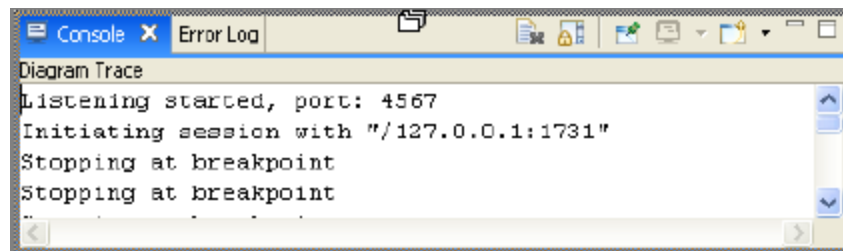
Rearrange a Group of Tabbed Views

You can rearrange a group of tabbed views in another location in the main window.

Follow these steps:

1. Right-click the title bar of one of the views in the tabbed group and select Move, Tab Group.

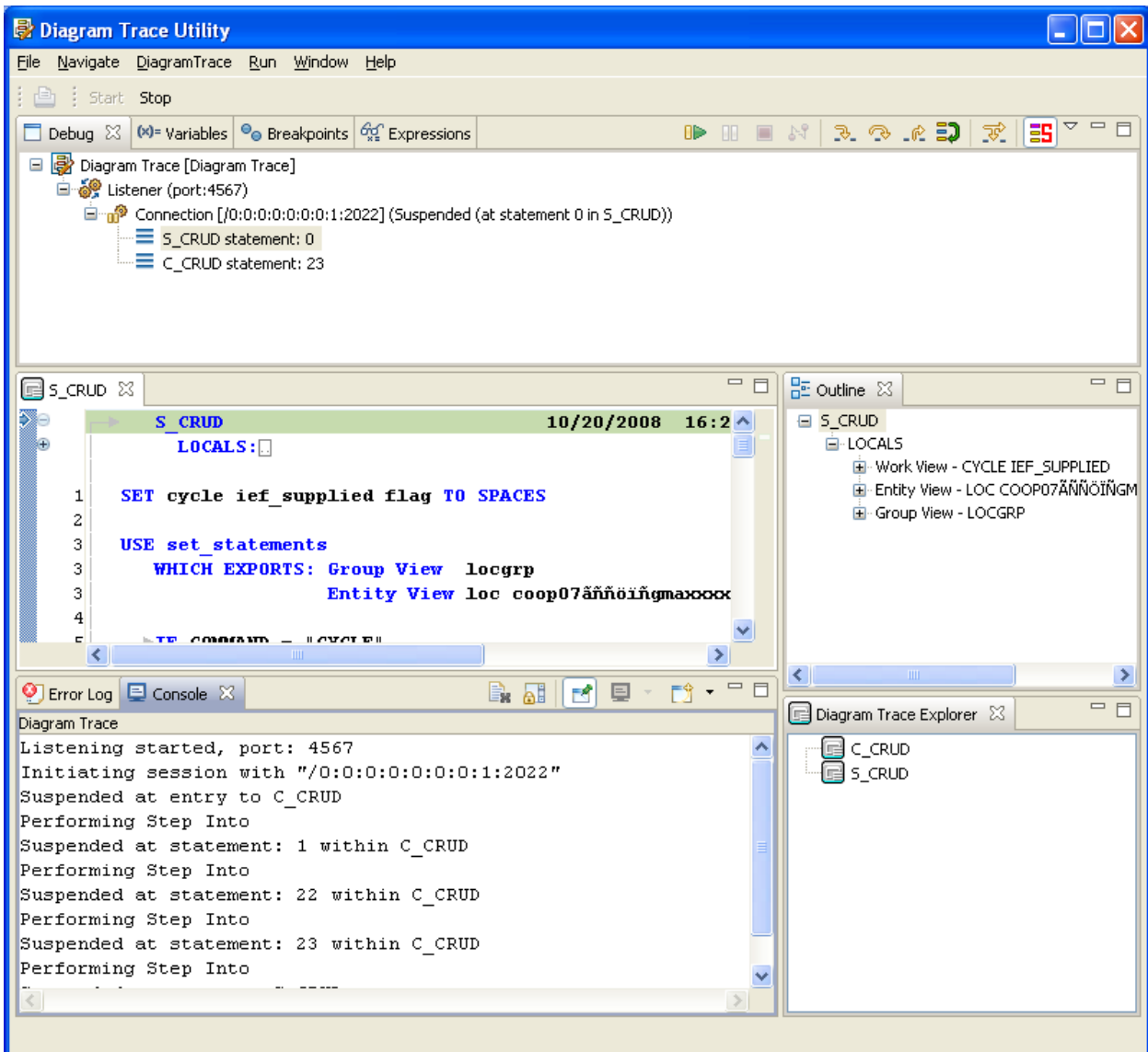
The tabbed group becomes dockable and the cursor changes to a stacked cursor as shown in the following illustration.



2. Move the group of tabbed views to another location and click the mouse once.

The group of tabbed views displays in the new location.

Example: The following illustration shows the Console and Error Log group of tabbed views rearranged below the PAD Code View editor.



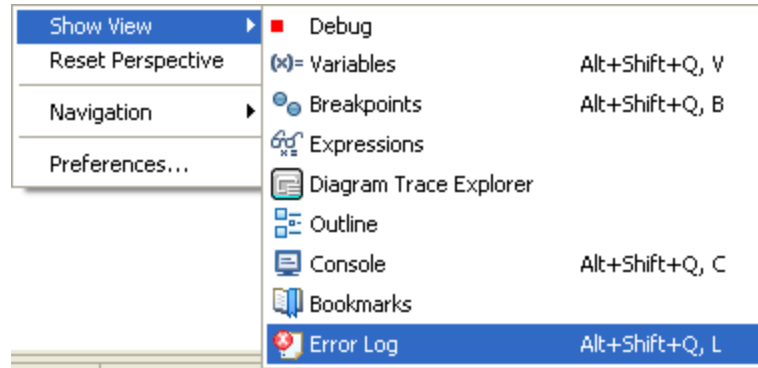
Open a View

You can open an undisplayed view as needed.

Follow these steps:

1. From the Windows menu, select Show View.

A list of views is displayed as shown in the following illustration:



2. Click the view you want to open.

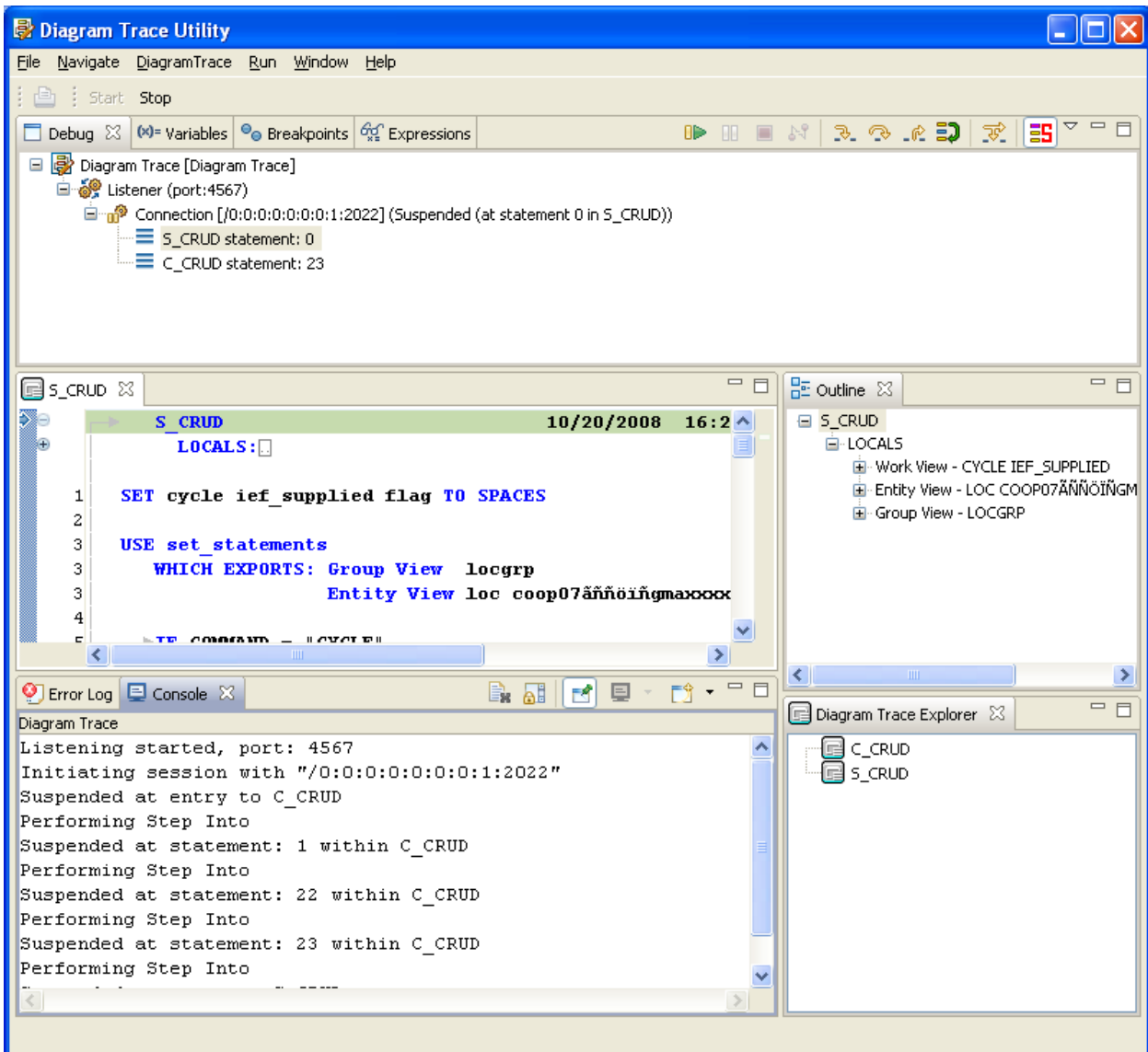
The selected view is displayed in the main window.

Note: If you use the Alt+Shift+Q key sequence to open a view instead of selecting the Show View option from the Windows menu, a pop-up menu appears that lets you select and open the following views:

- Breakpoints—Alt+Shift+Q, B
- Console—Alt+Shift+Q, C
- Error Log—Alt+Shift+Q, L
- Show View (View: Outline)—Alt+Shift+Q, O
- Variables—Alt+Shift+Q, V

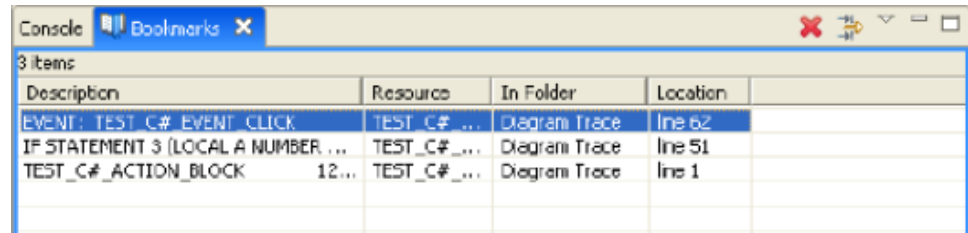
Note: The Diagram Trace utility does not support the Problems view.

Example: If you select Console, the utility displays the Console view in the Diagram Trace Utility main window. The following illustration shows the Console view added to the bottom of the main window. The Console view tab is highlighted, indicating that the view is the active view.



The Bookmarks View

The Bookmarks view displays all anchors (bookmarks) placed on a specific line of PAD code. The following illustration shows a sample Bookmarks view.



The screenshot shows a window titled 'Console' with a sub-tab 'Bookmarks'. It displays a table with 3 items. The table has four columns: Description, Resource, In Folder, and Location. The first row is highlighted in blue.

Description	Resource	In Folder	Location
EVENT: TEST_C#_EVENT_CLICK	TEST_C#_...	Diagram Trace	Line 62
IF STATEMENT 3 (LOCAL A NUMBER ...	TEST_C#_...	Diagram Trace	Line 51
TEST_C#_ACTION_BLOCK 12...	TEST_C#_...	Diagram Trace	Line 1

The bookmarks appear in a table format that provides a description of each bookmark, the name of the bookmark's resource, folder, and line number (location) in the PAD Code View editor.




Note: This view is not displayed in the default perspective. You must manually open the view by selecting Window, Show View, and Bookmarks on the Diagram Trace Utility main menu bar.

The Bookmarks View Toolbar

The Bookmarks view contains a toolbar that is displayed on the right side of the title bar.



The icons on the Bookmarks view toolbar let you perform the actions described in the following table:

Icon	Description
	Deletes a selected bookmark.
	Displays the Bookmark Filters dialog that lets you enable or disable bookmark filters on: <ul style="list-style-type: none">■ Any resource in the same project■ A selected resource only■ A selected resource and its children■ A selected Bookmarks working set
	Displays the Bookmarks Toolbar menu that contains options to: <ul style="list-style-type: none">■ Specify the sorting of displayed bookmarks (by Description, Resource, In Folder, Location, or Creation time) in either ascending or descending order■ Set Filters on bookmarks

The Bookmarks View Pop-up Menu

The Bookmarks view contains two pop-up menus that you can access by right-clicking in the view. The pop-up menu that displays depends on the location of your cursor in the view.

If your cursor is in the Description field of the Bookmarks view, a pop-up menu displays that lets you cut, copy, paste, or delete selected bookmarks.

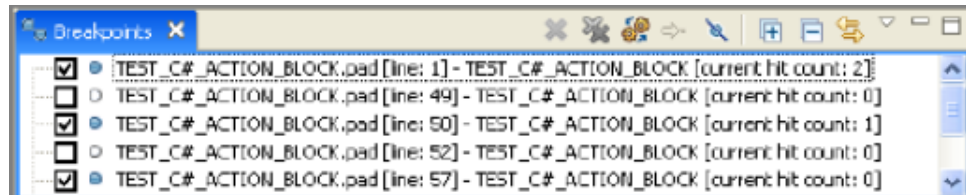
If your cursor is in the Resource column or the In Folder column of the view, a pop-up menu appears that lets you to perform the following actions:

- Go to the location of a selected bookmark in the PAD Code View editor.
- Copy one or more selected bookmarks.
- Delete a selected bookmark.
- Select all bookmarks.
- Display the Bookmarks Properties dialog for a selected bookmark.

Note: The Diagram Trace Utility does not support the Show in Navigator option on the Bookmarks view pop-up menu.

The Breakpoints View

The Breakpoints view displays all breakpoints (enabled and disabled) that are set in the vertical ruler of the PAD Code View editor.












Enabled Breakpoints appear as blue-filled circles to the left of the PAD code statements in the Breakpoints view and on the vertical ruler in the PAD Code View editor. Disabled breakpoints appear as white (unfilled) circles.

The Breakpoints View Toolbar

The Breakpoints View contains a toolbar that is displayed on the right side of the title bar.



The icons on the Breakpoints view toolbar let you perform the actions described in the following table:

Icon	Description
	Removes selected breakpoints
	Removes all breakpoints
	Shows breakpoints supported by selected target
	Goes to the file (in the PAD Code View editor) for a breakpoint
	Skips all breakpoints
	Expands all breakpoints in the Breakpoints view tree structure
	Collapses all breakpoints in the Breakpoints view tree structure
	Links to the Debug view
	Provides access to the Breakpoints Toolbar menu

The Breakpoints View Toolbar Menu

The Breakpoints View Toolbar menu commands let you perform the following actions:

Group By

Displays the following options that you can select:

- Breakpoints—Changes the Breakpoints view to show a simple list of all breakpoints with no groupings (the default view).
- Breakpoint Types—Changes the Breakpoints view to group the breakpoints by breakpoint type. There are three types of breakpoints: Line, Method, and Watchpoint.

- Breakpoint Working Sets—Changes the Breakpoints view to group the breakpoints by working sets.
- Files—Changes the Breakpoints view to group the breakpoints by the files or Action Diagrams to which they belong.
- Projects—Changes the Breakpoints view to group the breakpoints by the projects to which they belong.
- Resource Working Sets—Changes the Breakpoints view to group breakpoints by the resource working sets to which they belong.
- Advanced—Opens a dialog that lets you create a custom view that combines the other view types to create other breakpoint listings.

Select Default Working Set

Opens a dialog that lets you select (or create) a working set to be the default working set.

Deselect Default Working Set

Resets the view, so that there is no default working set. This option is enabled only if a default working set is defined.

Working Sets

Opens a dialog in which you can create, edit, or delete one or more working sets.

The Breakpoints View Pop-up Menu

You can access the pop-up menu by right-clicking in the viewing area of the Breakpoints view. This pop-up menu lets you perform the actions described in the following list:

Go to File

Switches to the breakpoint location in the PAD Code View editor, opening it if necessary.

Note: This option is only enabled when a line or method breakpoint is selected.

Enable

Sets the selected breakpoint to the Enabled state, places a check mark in the check box next to the breakpoint, and changes the breakpoint's icon on the vertical ruler in the PAD Code View editor to the enabled icon.

Note: If the breakpoint is already enabled, this option is not available.

Disable

Sets the selected breakpoint to the Disabled state, clears the check box next to the breakpoint, and changes the breakpoint's icon on the vertical ruler in the PAD Code View editor to the disabled icon.

Note: If the breakpoint is already disabled, this option is not available.

Remove

Removes one or more selected breakpoints from the Breakpoints view and removes the breakpoint icons from the vertical ruler in the PAD Code View editor.

Remove All

Removes all breakpoints from the Breakpoints view and removes all breakpoint icons from the vertical ruler in the PAD Code View editor.

Select All

Selects all of the breakpoints listed.

Note: This option is displayed and enabled only if at least one breakpoint is defined.

Copy

Copies the selected breakpoint to the clipboard.

Note: This option is only enabled when a breakpoint is selected.

Paste

Pastes the breakpoint to the selected group.

Note: This option is enabled only if you copied a breakpoint and enabled a group.

Properties

Opens the Breakpoint Properties page for the selected breakpoint.

Note: This option is only enabled when a breakpoint is selected.

Import Breakpoints

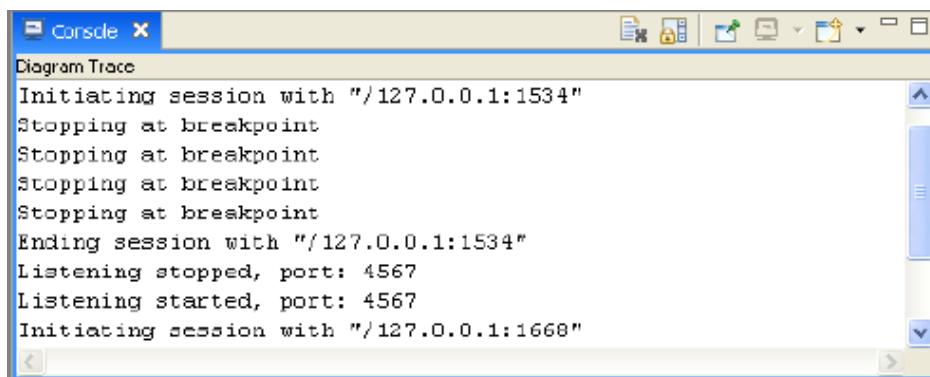
Opens the Import Breakpoints wizard which will help you import breakpoints into your workspace.

Export Breakpoints

Opens the export breakpoints wizard which will help you import breakpoints to a file.

The Console View

The Console view is a read-only view that displays standard output messages from the Diagram Trace Utility.






The Console View Toolbar

The following toolbar appears to the right of the Console view title bar.



The icons on the Console view toolbar let you perform the actions described in the following table:

Icon	Description
	Clears the contents of the Console view.
	Turns off automatic scrolling and locks the scroll bar where you placed it in the Console view.
	Provides a drop-down list of all open consoles. Only one console is shown and selecting it opens another detached view of the Console.

Note: The Diagram Trace Utility does not support the Display Selected Console and Pin Console icons on the Console view toolbar.

The Console View Pop-up Menu

You can access the Console view's pop-up menu by right-clicking in the viewing area of the Console view. The ConsoleView pop-up menu lets you perform the actions described in the following list:

Select All

Selects and highlights all text in the Console view.

Find/Replace

Finds and displays a specified word or phrase.

Note: The Diagram Trace Utility does not support the Replace action.

Clear

Clears the contents of the Console view.

Scroll Lock

Locks the Scroll bar where you placed it in the Console view.

Note: The Diagram Trace Utility does not support the Cut, Copy, Paste, or Open Link functions on the Console view pop-up menu.

More information:

[Review Contents of the Console Log View](#) (see page 139)

The Debug View

The Debug view lets you manage the debugging of an application. The view displays the current connection status for all configured ports and the execution status for each target application that is currently connected.

Debug View Artifacts and States

The following debug artifacts are displayed in the Debug view:

- The Diagram Trace Server
- At least one Listener process (default port is 4567)
- One or more Connections
- Stack Frames

These artifacts are displayed in a tree structure that follows the following conventions:

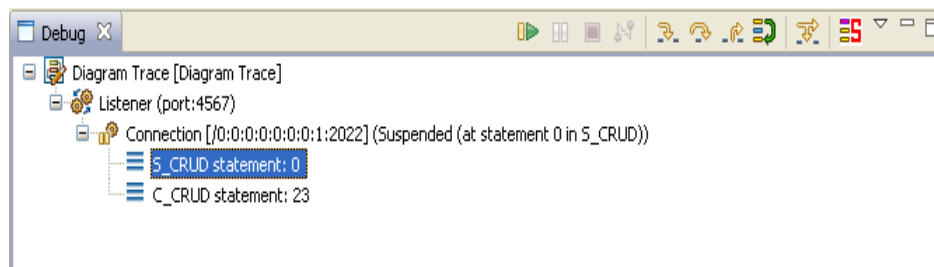
- The Diagram Trace Server is the root element
- Listener processes are children of the Diagram Trace Server
- Connections are children of a Listener process
- Stack Frames are children of a Connection
- All artifacts except Stack Frames can contain children

Artifacts in the Debug view can be in different states. You can change the state of an artifact using the enabled commands for that artifact as described in the following table:

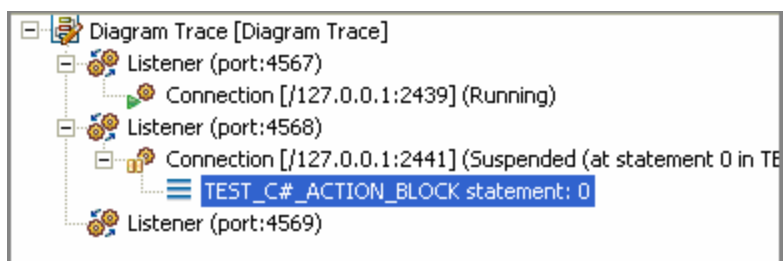
Artifact	States	Commands
Diagram Trace Server	Executing, Terminated	The Disconnect and Terminate commands change the state of the executing artifact to Terminated and removes the artifact.
Listener Process	Executing, Disconnected	The Disconnect command changes the state of the executing artifact to Disconnected and removes the artifact.
Connection	Running, Suspended	<p>The Suspend command changes the state of a running connection to Suspended. The notation <i><Suspended></i> appears to the right of the artifact's name stating the location in the PAD code that execution is suspended.</p> <p>The Resume command changes the state to Running. The notation <i><Running></i> appears to the right of the artifact's name.</p> <p>The Disconnect command removes this artifact and its children from the Debug view.</p>
Stack Frame	Suspended	<p>The Suspend command changes the state of an executing artifact to Suspended.</p> <p>The Resume command resumes execution of the Stack Frame.</p>

Sample Debug Views

The following sample Debug view shows that only the default Listener port (4567) is configured. The application is connected but is currently suspended.



You can configure multiple Listener ports. In the following example, three Listener ports (4567, 4568, and 4569) have been configured. The application connected to port 4567 is connected and is currently running. The application connected to port 4568 is suspended at statement 0 (entry to the Action Diagram). The configured port 4569 is listening for application connections, but currently has no connections.



Debug View Interactions with Other Views

Selecting actions in the Debug view causes the changes described in the following table to occur in other views.











Action	Resulting View Changes
Select a Single Stack Frame	<ul style="list-style-type: none"> ■ A PAD Code View editor for the Action Block is brought to the front of the editors (a new editor is opened if necessary). ■ When the front editor changes, the Outline view displays the outline for the selected Stack Frame. ■ The Variables view refreshes its display with the variables for the selected Stack Frame. ■ The Expressions view re-evaluates all expressions based on the selected Stack Frame. ■ The Breakpoints view highlights the Breakpoint related to the stopped state of the Stack Frame if the Breakpoints view has the Link With Debug View setting on.
Select a single Diagram Trace Server, Connection, or Listener process	<ul style="list-style-type: none"> ■ The Variables view resets its display to an empty view. ■ The Expressions view refreshes its display to no evaluations being performed (no Stack Frame context).
Selecting more than one item	<ul style="list-style-type: none"> ■ The Variables view resets its display to an empty view. ■ The Expressions view refreshes its display to no evaluations being performed (no Stack Frame context).

The Debug View Toolbar

The following toolbar appears to the right of the Debug view title bar.



The icons on the Debug view toolbar let you perform the actions described in the following table:

Icon	Action	Description
	Resume	Resumes execution on one or more selected connections.
	Suspend	Suspends execution on one or more selected connections.
	Terminate	Terminates the selected debug session.
	Disconnect	Disconnects from one or more selected connections.
	Step Into	Resumes execution, stopping at the next statement either in the same method, a called method or (if you are at the end of a method) in the method from which the current method was called.
	Step Over	Resumes execution, stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.
	Step Return	Resumes execution, stopping at the next statement in the method from which the current method was called.
	Skip Over	Resumes the application (without executing the current statement), stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.
	Use Step Filters	Toggles step filters on and off. When on, all Step actions apply step filters.
	Suspend on Entry	Suspends the application execution whenever an Action Diagram is first entered.

To determine the action of each icon, hover your mouse over the icon to see the tooltip that appears.

The Debug View Pop-up Menu

You can access the Debug view's pop-up menu shown in the following illustration by right-clicking inside the Debug view. The Debug View pop-up menu lets you perform the actions described in the following list:

Step Into

Resumes execution, stopping at the next statement either in the same method, a called method or (if you are at the end of a method), in the method from which the current method was called.

Step Over

Resumes execution, stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Step Return

Resumes execution, stopping at the next statement in the method from which the current method was called.

Skip Over

Resumes the application (without executing the current method), stopping at the statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Use Step Filters

Toggles the Use Step Filters action to turn on the use of step filters (so all Step actions apply step filters) or turn off the use of step filters.

Resume

Resumes execution on one or more selected connections.

Suspend

Suspends execution on one or more selected connections when the selections contain only running Connections or Stack Frames.

Terminate

Terminates the selected debug target and ends the debug session.

Disconnect

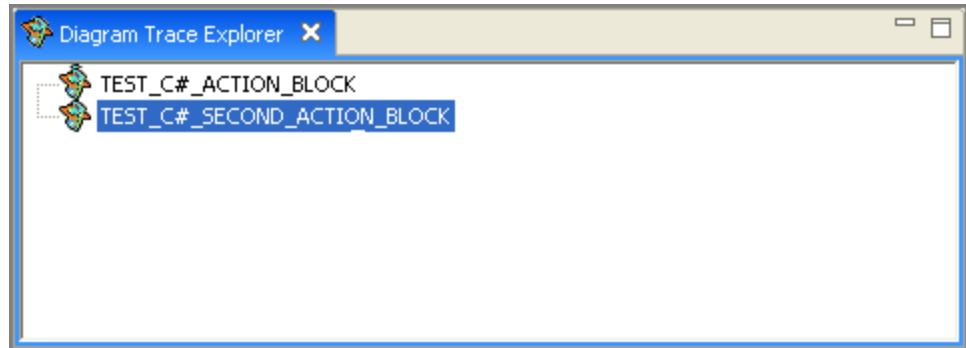
Disconnects from the selected application, which effectively terminates the debug session.

Note: The Diagram Trace Utility does not support the Properties function on the Debug view pop-up menu.

The Diagram Trace Explorer View

The Diagram Trace Explorer view displays a hierarchical view of the source code for PADs that are cached in the Diagram Trace Utility workbench.

The illustration shows two cached Actions Diagrams that are available for inspection. You can right-click and select Open from the pop-up menu to open the available source (.pad) files inside of a PAD Code View editor.



During the debug session, the application sends its PAD code to the Diagram Trace Utility. The Diagram Trace Utility stores the received data in XML format in .pad files in the Diagram Trace Utility's workspace subdirectory. The utility saves these files from one debug session to another and lets you access these .pad files.

The Diagram Trace Explorer View Pop-up Menu

You can access the pop-up menu in the Diagram Trace Explorer view by right-clicking inside the view. The Diagram Trace Explorer pop-up menu lets you perform the actions described in the following list:

Open

Opens the selected Action Diagram in the PAD Code View editor.

Delete

Deletes the .pad file containing the selected Action Diagram, which has the following effects:

- The deleted Action Block is removed from the list in the Diagram Trace Explorer.
- All breakpoints associated with the deleted Action Block are removed.
- All bookmarks associated with the deleted Action Block are removed.
- Any open PAD Code View editors for the deleted Action Block are closed.

Select All

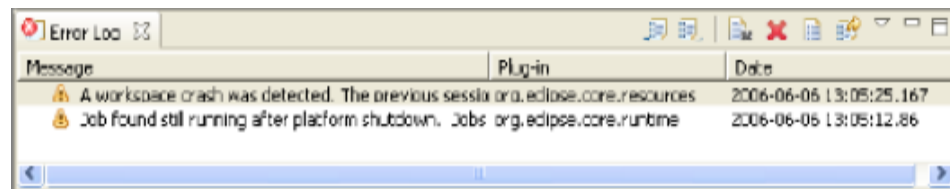
Selects all listed Action Diagrams.

Refresh

Refreshes or updates the view's data area.

The Error Log View

The Error Log view captures and displays all internal warnings and errors generated by the platform and your code. The Diagram Trace Utility writes these errors to a .log file created in the .metadata subdirectory of your workspace. The Error Log view shows the content of this file and provides options to sort or filter your data.










By default, after the Error Log view is in your perspective, the Diagram Trace Utility activates (highlights) the view when new events are logged. You can turn this feature on or off using the view's drop-down menu.

The Error Log View Toolbar

The following toolbar appears to the right of the Error Log view title bar.



The icons on the Error Log view toolbar let you perform the actions described in the following table:

Icon	Description
	Exports the Error log displayed in the Error Log view to a specified file on your local system or network drive.
	Imports a specified Error log from your local system or network drive.
	Clears the current contents of the Error Log view (the Error Log file remains intact).
	Deletes the Error Log from the Error Log view.
	Opens the Error log detail in Notepad.
	Restores the contents of the Error Log view.
	Displays the Toolbar menu that contains options to: <ul style="list-style-type: none">■ Set Filters for the type of error messages (Information, Warning, or Error) displayed in the Error View log and specifies whether to display the events logged during all sessions or during the most recent session.■ Activate the log when new events occur.

The Error Log Pop-up Menu

The Error Log view contains a pop-up menu, which you can access by right-clicking in the view. The pop-up menu lets you perform the actions described in the following list:

Copy

Copies the item currently selected to the Windows clipboard.

Clear Log Viewer

Clears the log data displayed in the Error Log view (the Error Log file remains intact).

Delete Log

Deletes the contents of the Error log.

Open Log

Opens the current Error log using Notepad to view Error Log content that may not be displayed in the Error Log view when the view was cleared.

Restore Log

Restores the entire contents of the current Error Log to the Error Log view display.

Export Log

Exports the current Error Log file to an external file on the Windows file system.

Import Log

Imports an Error Log you exported from a previous debug session to replace the current Error Log file. The imported file must adhere to the Error Log file format to be displayed correctly in the Error Log view. The imported Error Log is for viewing purposes only.

Note: Error output from the current debug session continues to be written to the Error Log file that was in use when you started the Diagram Trace Utility. The imported Log currently being viewed is not updated with errors pertaining to the current debug session.

If you import an Error Log file, the text of the Error Log view toolbar icon and the Pop-up menu option used to restore the current log changes from Restore Log to Restore This Workspace's Log. The Restore This Workspace's Log function causes the current debug session Error Log file to replace the previously imported log file in the Error Log view.

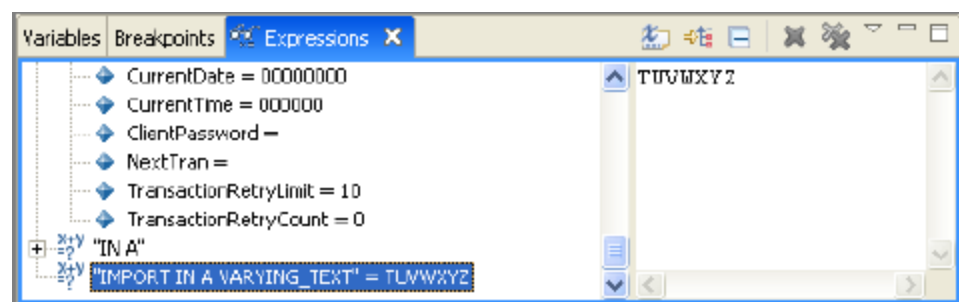
Event Details

Opens an Event details dialog to display a detailed explanation of the selected error.

The Expressions View

In the Expressions view, you can view the current evaluated state of expressions that you previously entered.

The Expressions view opens automatically when you add a Watch expression to the view. You can inspect data in the Expressions view when the trace execution is suspended.



There are two split view panes displayed by default in the Expression view, the Expression pane and the Detail pane. The Expression pane contains the entered or copied expression to be evaluated. The Detail pane displays the values resulting from the highlighted expression or expressions that are being evaluated.

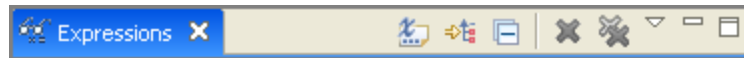
When the Detail pane is displayed, you can click the *slider bar* that exists between the two areas and move it to adjust the size of each pane.

More information:






[Expressions](#) (see page 14)

The Expressions View Toolbar

The Expressions View contains the following toolbar:



The icons on the Expressions view toolbar let you perform the actions described in the following table:

Icon	Description
	Toggles the display of the variable type names and sizes that precede the variable name in the Expression pane.
	Collapses the expanded tree view.
	Removes selected expressions from the Expressions view.
	Removes all expressions from the Expressions view.
	Displays the Details toolbar menu.

Note: The Diagram Trace Utility does not support the Show Logical structure icon on the Expressions view toolbar.

Set the Display Orientation of the Expressions Panes

You can set the display orientation of the Expressions panes using the Layout option under Menu on the right side of the Expressions view title bar. To set the Display orientation of the Expressions panes, select one of the following menu options from Menu, Layout.

Vertical View Orientation

Aligns the Variables pane to appear above the Detail pane

Horizontal View Orientation

Aligns the Variables pane to appear to the left of the Detail pane

Expression View Only

Removes the Detail pane from the display

You can drill down the repeated group values and hide the non active display groups by selecting the following options under Menu in the Expressions view title bar.

Expand RGV Breadth First

Drills down through occurrences of repeated group values to the bottom attributes and displays them as arrays of values instead of records

Hide Non-active Group View Rows

Hides the display group rows that are not active currently

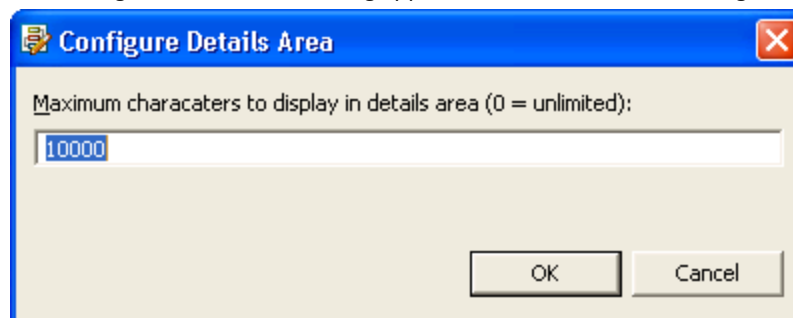
Configure the Maximum Characters to Display in the Expressions Detail Pane

You can configure the maximum number of characters to display in the Expression Details pane in the Expression view.

Follow these steps:

1. Right-click on the Details pane in the Expression view.

The Configure Details Area dialog appears as shown in the following illustration:



2. Replace the default entry (10000) with the maximum number of characters to display in the Details pane and click OK.

The configuration is saved and the dialog closes.

Enable and Disable Text Wrapping in the Expression Details Pane

Text wrapping in the Expressions Detail pane is disabled by default.

To enable and disable text wrapping in the Expressions Detail pane, right-click on the Details area and select the option. If text wrapping is enabled, selecting the option disables text wrapping. If text wrapping is disabled, selecting the option enables text wrapping.

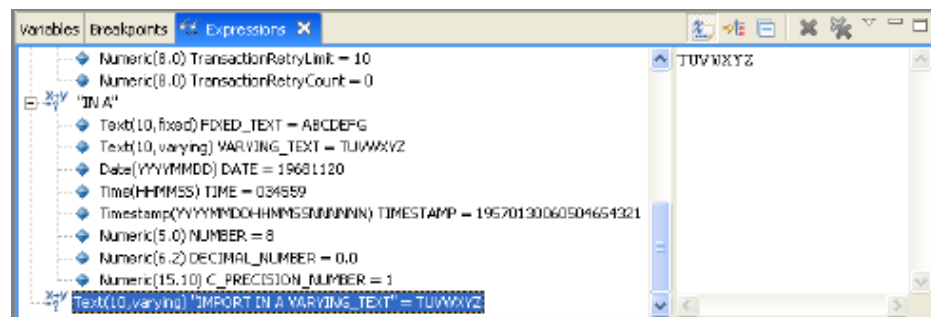
Add a Watch on Selected Variable Views

The Diagram Trace Utility lets you enable a Watch on selected variables (views or attributes) in the Variables view or the Outline view. To add a watch on selected variables, select one or more variables in the Variables view or the Outline view, right-click and select Watch from the pop-up menu.

The Diagram Trace utility adds a new Watch expression to the Expressions view for each variable view you selected in the Outline view and brings the Expressions view to the front of the display (or opens the view if it is not displayed).

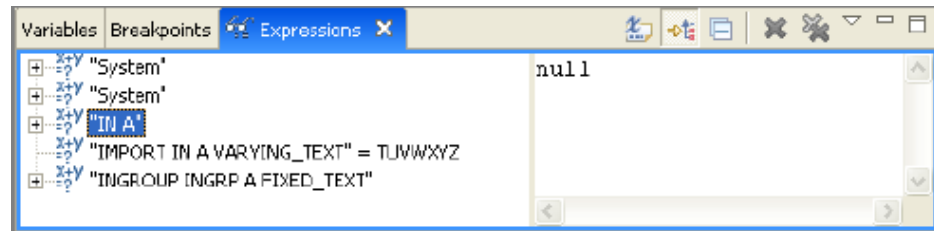
Show Variable Type Names

Click the Show Type Names icon on the Expressions view toolbar to display or hide variable type names and sizes that precede the variable name in the Expressions pane of the Expressions view. The following illustration displays the variable type names:



Collapse the Expanded Tree Structure

Click CollapseAll in the Expression view toolbar to collapse the expanded tree view in the Expressions view. The following illustration shows a collapsed tree view:



Remove a Selected Expression

You can select one or more expressions to be removed from the Expressions view.

Follow these steps:

1. Click one or more expressions to be removed.

Note: You can select multiple expressions by holding down the Ctrl key and clicking on the expressions you want to remove.

The expressions to be removed are selected.

2. Click Remove Selected Expressions on the Expressions view toolbar.

The expressions are removed from the Expressions view.

Remove All Expressions

You can remove all expressions that have been added to the Expressions view. To remove all expressions from the Expressions view, click Remove All Expressions on the Expressions view toolbar.

The Expressions View Pop-up Menu

You can access the Expressions view pop-up menu by right-clicking inside the Expressions view. The Expressions view pop-up menu lets you perform the following actions:

Select All

Selects all expressions in the Expressions view

Copy Expressions

Copies selected expressions to the clipboard

Remove

Removes a selected expression from the Expressions view

Remove All

Removes all expressions from the Expressions view

Find

Finds all the expressions in the Expressions view

Change Value

Displays the Change Expression Value dialog and lets you enter a value for a selected expression

Add Watch Expression

Adds a Watch expression to the Expressions view

Reevaluate Watch Expression

Re-evaluates a Watch expression

Disable

Disables a Watch expression in the Expressions view and adds the annotation (*disabled*) to the right of the expression

Enable

Enables a Watch expression that was previously disabled

Edit Watch Expression

Opens the Edit Watch Expression dialog and lets you edit the text of a Watch expression

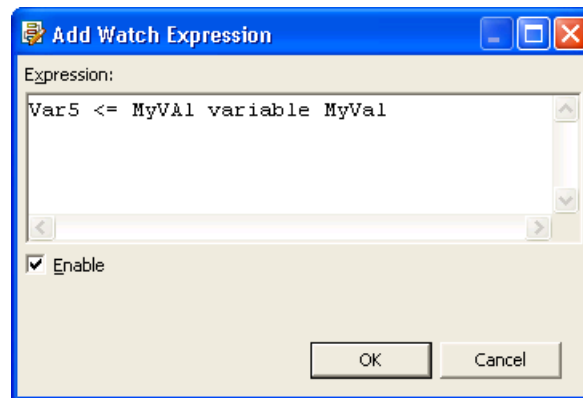
Add a Watch Expression

You can add one or more Watch expressions to the Expressions view using the view's pop-up menu.

Follow these steps:

1. Right-click in the Expressions view and select Add Watch Expression from the pop-up menu.

The Add Watch Expression dialog appears.



2. Clear the Enable check box if you want the Watch expression to be disabled, or select the check box to enable the Watch expression and click OK.

The dialog closes and the Watch expression is added to the Expressions view.

Edit a Watch Expression

After you have added one or more Watch expressions to the Expressions view, you can edit the Watch expressions using the pop-menu menu.

Follow these steps:

1. Select the Watch expression to be edited, right-click and select Edit Watch Expression from the pop-up menu.

The Edit Watch Expression dialog appears and displays the Watch expression you selected to be edited.

2. Make the necessary edits and click OK.

The edited Watch expression is saved in the Expressions view.

Enable or Disable a Watch Expression

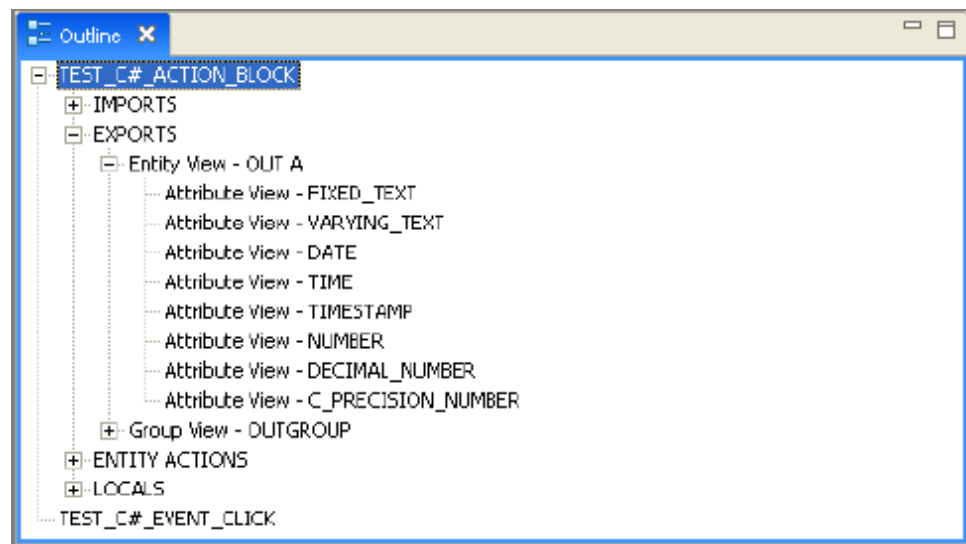
You can disable or enable one or more Watch expressions as necessary through the Expressions View pop-up menu.

To enable a Watch expression, right-click on a disabled Watch expression and select Enable from the pop-up menu. The (disabled) annotation is removed from the Watch expression, indicating that it is enabled.

To disable a Watch expression, right-click on an enabled Watch expression and select **Disable** from the pop-up menu. The (disabled) annotation appears to the right of the selected Watch expression, indicating that it is disabled.

The Outline View

The Outline view displays an outline of the application PAD file that is currently open in the PAD Code View editor area and lists the structural elements. The contents of the Outline view are specific to the PAD file.



The Outline view contains a top level node for each method (action block or any event actions defined in the action block) in the action block. Each method node expands into nodes for each of its attribute views, base views, group views, entity views, and work set views.

Selecting a node switches to the location of that node in the PAD Code View editor. If the location is collapsed, the area is automatically expanded.

Note: If you activate a new PAD Code View editor, the contents of the Outline view switch to the outline for the new editor. The expanded state of the outline is preserved as long as the PAD Code View editor with which the outline is associated remains open. If you close the Outline view and open it again, the expanded state of the view nodes are not preserved.

The Outline View Pop-up Menu

You can access the Outline view pop-up menu options by selecting an item in the view and right-clicking your mouse. The Outline view pop-up menu options that appear depends on the type of item you select:

Toggle Method Breakpoint

Appears if you select one or more methods. This option either adds a method breakpoint to the Breakpoints view for each selected method (if no breakpoint exists) or removes an existing breakpoint from the Breakpoints view for each selected method.

Watch

Appears if you select one or more views. This option adds a Watch expression to the Expressions view for each selected view.

Toggle Watchpoint

Appears if you selected one or more variable views. This option adds a watchpoint (data breakpoint) to the Breakpoints view for each selected variable view or removes an existing watchpoint for each selected variable view.

Toggle a Method Breakpoint in the Outline View

You can toggle (add or remove) a method breakpoint in the Outline view. The breakpoints appear in or disappear from the Breakpoints view, depending on the action that was taken. New breakpoint icons appear on the vertical ruler of the PAD Code View editor.

To toggle a method breakpoint, select one or more methods in the Outline view, right-click, and select Toggle Method Breakpoint from the pop-up menu.

The Diagram Trace Utility takes one of the following actions:

- Adds a method breakpoint for each selected method when no method breakpoint exists
- Deletes a method breakpoint for each selected method when a method breakpoint exists

Toggle a Watchpoint in the Outline View

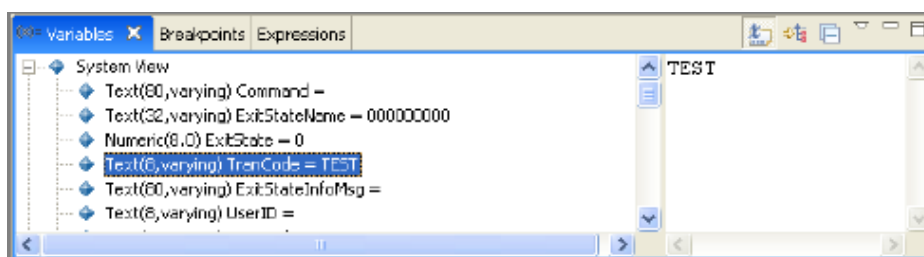
The Diagram Trace Utility lets you toggle a watchpoint (data breakpoint) on selected variable views using the pop-up menu option Toggle Watchpoint in the Outline view. This toggle adds a data breakpoint to the Breakpoints view for each selected variable if no watchpoint exists, or removes the existing breakpoint associated with each selected variable.

The inherent condition associated with this data breakpoint is *break if value has changed*. The value of the selected variable view is evaluated before the execution of each statement. When the value changes, program execution is suspended at the current executable statement.

The Variables View

The Variables View displays information on the values of variables in a selected stack frame. The variables displayed in the view include CA Gen Action Diagram system variables and import, export, entity, and local view data for all views currently in scope. The variables are organized in a tree structure that lets you move down to the lowest level object value that is displayed. You can expand and collapse the tree structure as needed.

The following sample Variables View shows the names of the different types of variables in the Variable Name pane and shows the value of the selected variable in the optional Value Detail Pane. You can select multiple variables.



In the Variable Name pane, you can inspect and modify CA Gen Action Diagram system variables and add variable values to the Expressions view. In addition, you can add watchpoints and can toggle watchpoints to enable breakpoints that are based on the changing values of variables.

Note: The value of the BLOB attribute is not shown in the Variables View. The following format represents the BLOB attribute value:

```
{BLOB;maxLength=nnnn;length=nnnn}
```

BLOB

Specifies the BLOB variable.

maxLength

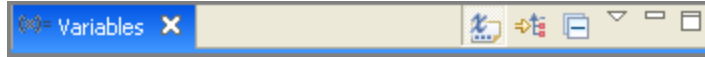
Specifies the maximum length of the BLOB variable.

Length




Specifies the actual length of the BLOB variable.

The Variables View Toolbar

The Variables view contains the following toolbar on the right side of the Variables view title bar:



The icons on the Variables view toolbar let you perform the actions described in the following table:

Icon	Description
	Toggles the display of the variable type names and sizes that precede the variable name in the Variables pane.
	Collapses the expanded tree view.
	Displays the Detail pane toolbar menu.

Note: The Diagram Trace Utility does not support the Show Logical Structure icon on the Variables view toolbar.

Set the Display Orientation of the Variables Panes

You can set the display orientation of the Variables panes using the Menu on the right side of the Variables view title bar.

To set the Display orientation of the Variables panes, select one of the following Menu, Layout.

Vertical View Orientation

Aligns the Variables pane to appear above the Detail pane

Horizontal View Orientation

Aligns the Variables pane to appear to the left of the Detail pane

Variables View Only

Removes the Detail pane

Show Columns

Toggles whether the variables view displays information using table columns. (Only displayed when application is suspended.)

Select Columns...

Opens a dialog allowing the columns shown to be customized. (Only displayed when 'Show Columns' is checked and application is suspended.)

You can drill down the repeated group values and hide the non active display groups by selecting the following options under Menu in the Expressions view title bar.

Expand RGV Breadth First

Drills down through occurrences of repeated group values to the bottom attributes and displays them as arrays of values instead of records

Hide Non-active Group View Rows

Hides the display group rows that are not active currently.

Configure the Maximum Characters to Display in the Variables Detail Pane

You can configure the maximum number of characters to display in the Variables Details pane in the Variables view.

Follow these steps:

1. Right-click on the Details pane in the Variables view.
The Variables view menu options appear.
2. Select Max Length menu option.
The Configure Details Area dialog appears.
3. Replace the default entry (10000) with the maximum number of characters to display in the Details pane and click OK.
The configuration is saved and the dialog closes.

Variables View Pop-up Menu

The Variable view contains a pop-up menu that you can access by right-clicking in the view. The pop-up menu lets you perform the following actions:

Select All

Selects all variables displayed in the Variables view

Copy Variables

Copies one or more selected variables to the Windows clipboard

Find

Displays the Find Variable dialog and lets you type a variable to display or select a variable to display from a list of system variables in the Variables view

Change Value

Displays the Change Variable Value dialog and lets you enter a value for a selected system variable

Toggle Watchpoint

Adds a watchpoint (data breakpoint) to the Breakpoints view for each variable selected in the Variables view or removes an existing watchpoint from the Breakpoints view for each variable selected

Create Watch Expression

Adds a Watch expression to the Expressions view for each selected variable and brings the Expressions view to the front of the display (or opens the Expressions view if it is not displayed)

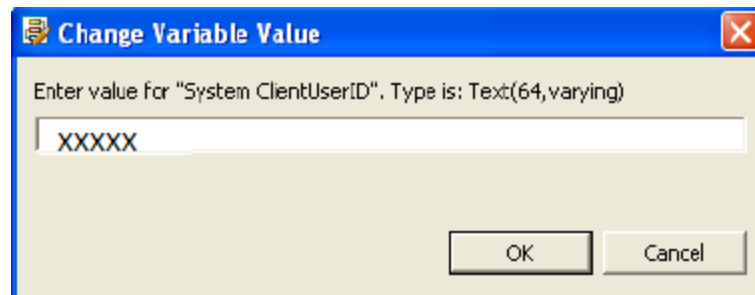
Change a Variable Value

You can change a value for a selected variable using the pop-up menu.

Follow these steps:

1. Select a variable name from the Variable Name list, right-click, and click Change Value from the pop-up menu.

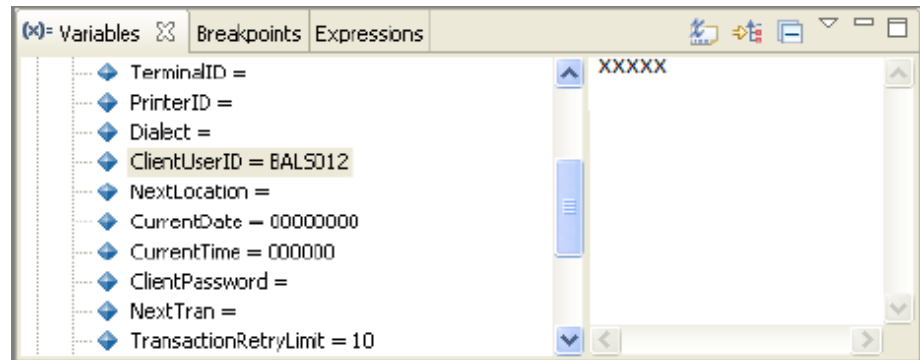
The Change Variable Value dialog appears.



2. Enter a value for the selected variable and click OK.

The new value is set for the variable.

Example: The following illustration shows the new value for ClientUserID in the Details pane.



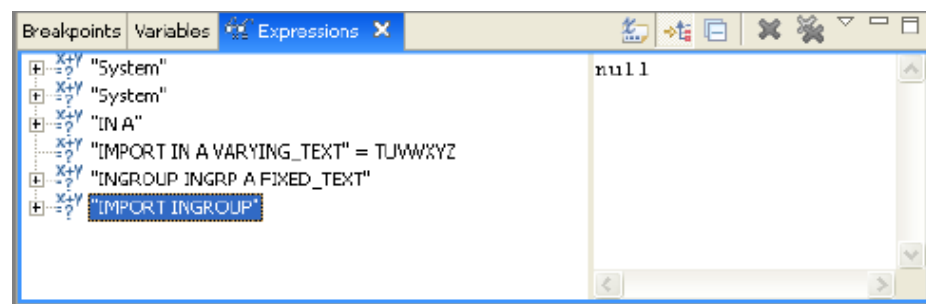
Note: The Diagram Trace Utility performs data validation (such as disallowing letters in numeric fields) as you type each character. The Diagram Trace Utility enforces time and date formats, size limits, and numeric precision limits. The utility does not change the value if the validation fails.

Note: You cannot modify BLOB data using Diagram Trace Utility.

Add a Watch on a Variable

To add a watch on a variable, select a variable name from the Variable Name list (for example, Group View—InGROUP), right-click, and select Create Watch Expression from the pop-up menu.

The Diagram Trace Utility adds an expression to the Expressions view for the selected variable and brings the Expressions view to the front of the display (or opens the view if it is not displayed).



Toggle a Watchpoint

The Diagram Trace Utility lets you toggle a watchpoint (data breakpoint) on one or more selected variables in the Variables view. This toggle adds a data breakpoint to the Breakpoints view for each selected variable if no watchpoint exists or removes the existing breakpoint associated with each selected variable.

The inherent condition associated with this data breakpoint is *break if value has changed*. The value of the selected variable is evaluated before the execution of each statement. When the value changes, program execution is suspended at the current executable statement.

To Toggle a Watchpoint, select a variable in the Variables view, right-click the selected variable and select Toggle Watchpoint from the pop-up menu.

PAD Code View Editors

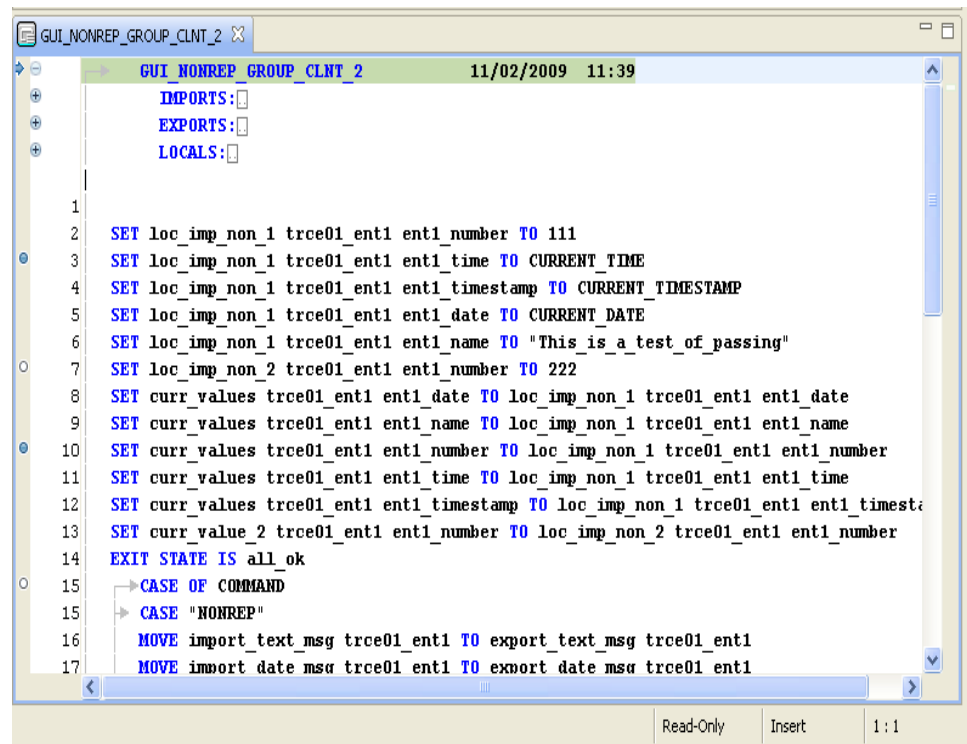
PAD Code View editors are visual components in the Diagram Trace Utility main window. The main menu bar and toolbar on the Diagram Trace window contain operations that are applicable to the active PAD Code View editor.

You can start an editor inside the PAD Code view area of the Diagram Trace Utility main window to view CA Gen PAD code received from the application being debugged. Any number of PAD Code View editors can be open at one time, but only one editor can be active at any given time.

Important! The PAD Code View editors in the Diagram Trace Utility are read-only viewers. You cannot modify the PAD code in these editors. You must make any modification to the model using the CA Gen Toolset.

Tabs in the PAD Code View editor area indicate the names of resources that are currently open for viewing. By default, editors are stacked in the PAD Code View editor area, but you can tile them to view Action Diagrams simultaneously. You can also minimize and maximize PAD Code View editors to increase or reduce the amount of window space they occupy.

The following illustration shows an example of a PAD Code View editor view.



If you have set one or more breakpoints for tracing the program, they are displayed as blue or white circles on the highlighted border at the left margin of the PAD Code View editor. Blue circles indicate enabled breakpoints and white circles indicate disabled breakpoints.

More information:

[Set Annotations Preferences for Text Editors](#) (see page 129)

The Vertical Ruler and Pop-up Menu

The vertical ruler is the vertical bar located at the left of the PAD Code View editor area. Right-click the vertical ruler and the pop-up menu appears. The following list describes the options on the vertical ruler pop-up menu:

Toggle Breakpoint

Enables or disables a breakpoint.

Enable/Disable Breakpoint

Changes the state of the current breakpoint from enabled to disabled or from disabled to enabled. The menu item text changes appropriately.

Breakpoint Properties

Displays the Properties dialog for the selected breakpoint. You can use this dialog to set or modify various breakpoint properties.

Add/Remove Bookmark

Adds or removes a bookmark for a selected line of PAD code.

Show Line Numbers

Enables or disables display of line numbers on the PAD code.

Show Statement Numbers

Enables or disables display of the CA Gen statement numbers.

Preferences

Opens Preferences dialog that lets you review or change the preferences associated with a PAD Code View editor.

Activate a PAD Code View Editor

You can activate multiple PAD Code View editors in the PAD Code view, but only one editor can be active at a time.

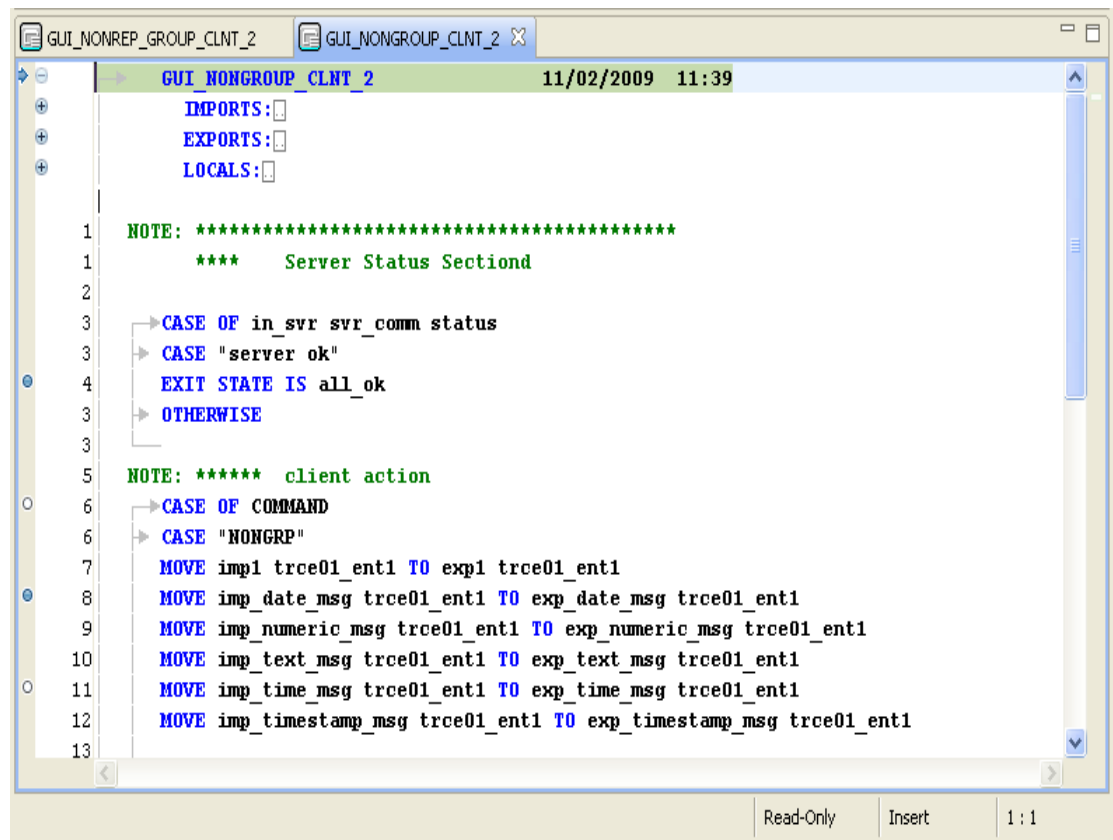
To activate a PAD Code View editor, click the tab of the editor you want to activate. The Diagram Trace Utility highlights the selected PAD Code View editor tab indicating that it is the active editor.

Add a New PAD Code View Editor

You can add one or more new PAD Code View editors through the pop-up menu on PAD Code viewing area.

To add a new PAD Code View editor, right-click on the title bar of the PAD Code View editor and select New Editor from the pop-up menu. The new editor appears in the PAD Code view with the default editor as a tabbed group.

Example: In the following illustration, the new PAD Code View editor is highlighted, indicating it is the active editor.



Tile PAD Code View Editors

You can create two or more sets of PAD Code View editors in the PAD view editor area. You can also resize the PAD Code View editor area, but you cannot drag views into the PAD Code View editor area.

Follow these steps:

1. Open at least two PAD Code View editors in the Editor area by double-clicking editable files in one of the navigation views and drag one of the editor's tabs out of the PAD Code View editor area.

Note: The *restricted* cursor displays if you try to drop the editor on top of any view or outside the Diagram Trace Utility window.

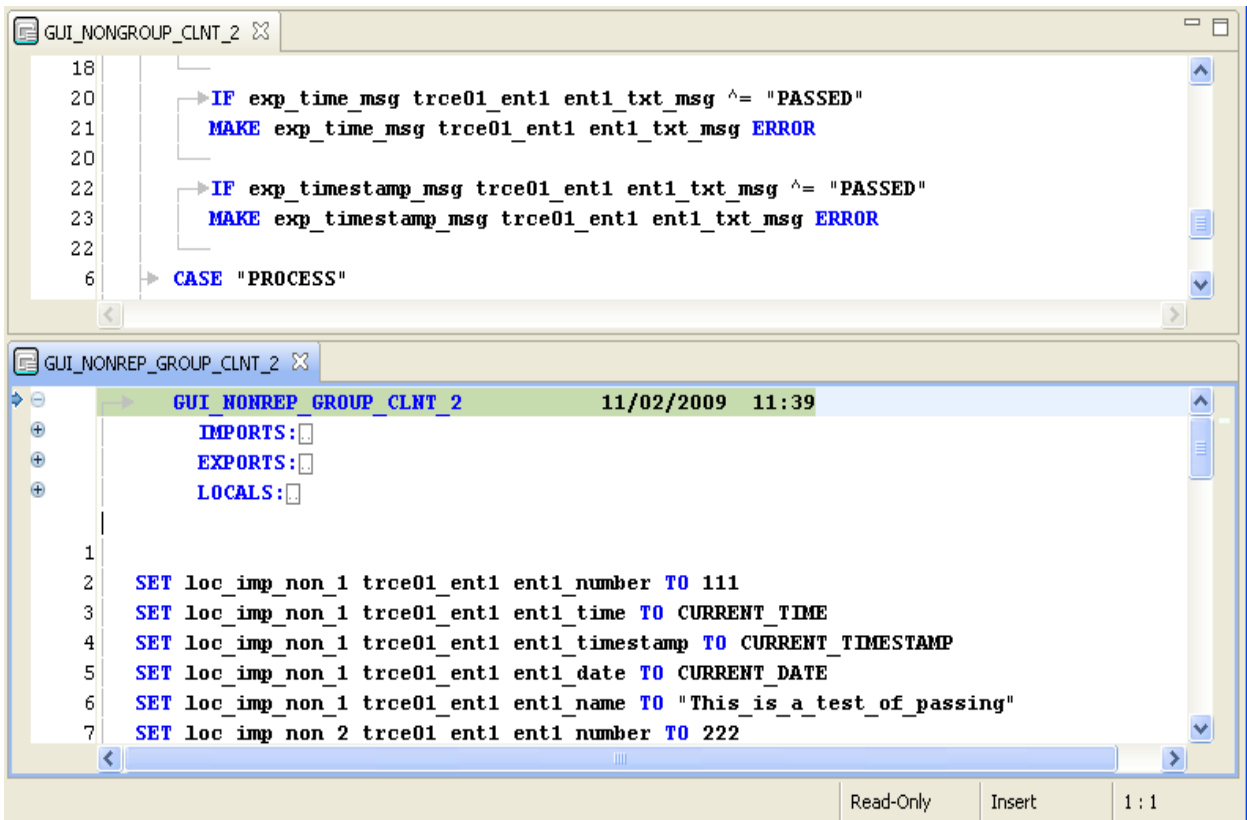
2. Dock the editor on a directional arrow drop cursor so that two editors appear in the editor area.

Note: You can resize each editor and the entire Editor area to accommodate the editors and views as necessary.

3. (Optional) Observe the color of the editor tabs.

- Blue—Indicates that the editor is currently active.
- Default color (depends on the system color scheme)—Indicates that the editor was the last active editor. If there is an active view, it is the editor that the active view is currently working with. This is important when working with views such as the Outline and Variables view that work closely with the editor.

Example: The following illustration shows a layout of tiled editors:



Rearrange the Order of Tabbed PAD Code View Editors

You can rearrange the order of a tabbed group of PAD Code View editors by dragging the title bar of one of the tabbed editors to the left or right of another editor and releasing the mouse button when the cursor appears as a Stack cursor. The PAD Code View editor you moved becomes the active editor.

Import Preferences or PAD Files into a Debug Session

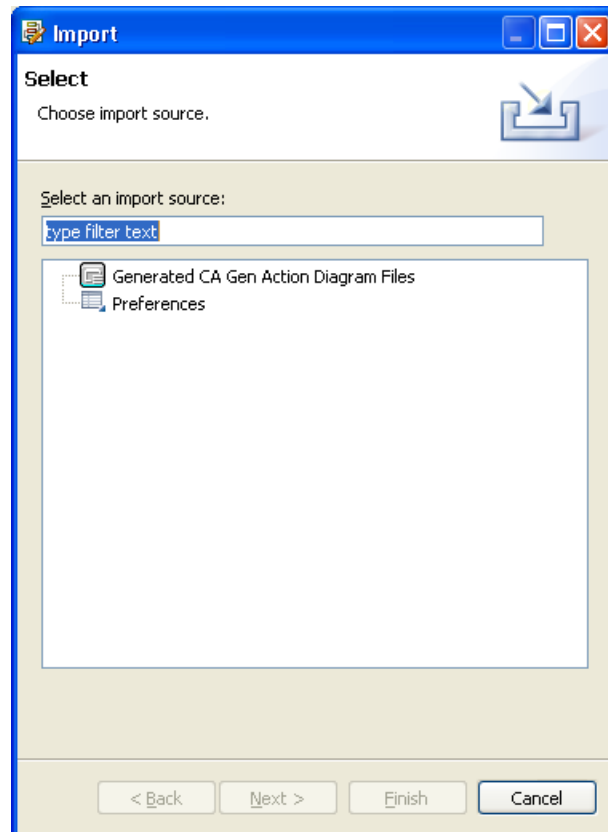
The Import Preferences feature provides a wizard that lets you import preference settings from a file using the following import options:

- Import preference settings from a previously exported Preference file saved on the local system.
- Import CA Gen PAD files from generated Action Diagram code in a generated target language source file. The file to be imported must be one of the following generated source file types:
 - .c or .sqc (C language)
 - .java (Java language)
 - .cs (C Sharp.NET language)

Follow these steps:

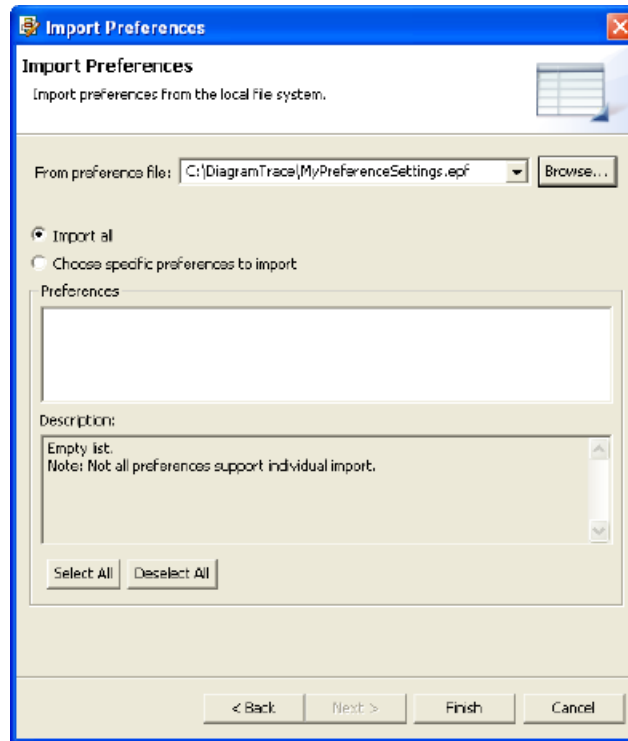
1. Select File, Import from the main menu.

The Select an Import Source dialog appears.



2. Click Preferences and click Next.

The Import Preferences dialog appears.



3. Click Browse to open the Windows Import from File dialog and select the file that contains the Diagram Trace Utility Preference settings you previously exported, and then click OK.

The path for the selected file appears in the From preference file field.

Note: The name of the specified preference import file must have an .epf suffix.

4. Select Import All or select specific preferences to import from the displayed selection list and click Finish.

The preferences you selected are imported into the current debugging session.

Note: The Diagram Trace Utility does not support the Choose specific preferences to import option.

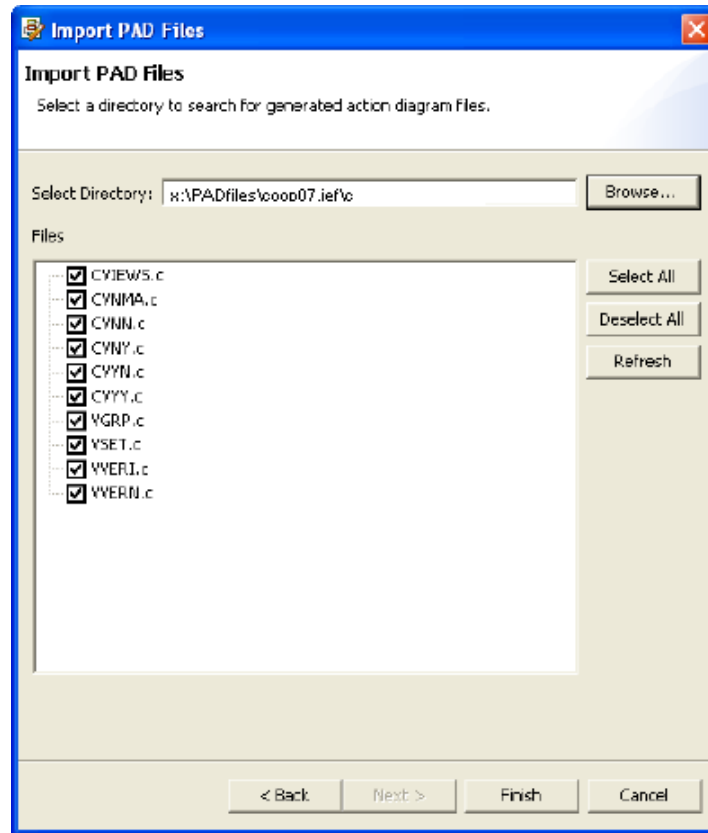
To import CA Gen PAD files from generated Action Diagram code

1. Select File, Import from the File menu on the main menu bar.

The Select an Import source dialog appears.

2. Select Generated CA Gen Action Diagram Files and click Next.

The Import PAD files dialog appears.



3. Click Browse to open the Window Browse for Folder dialog and select the folder containing the generated PAD files to import.

The path to the selected folder appears in the Select Directory: field.

4. Click Select All to select all PAD files and click Finish.

Note: If you do not want to import all files, you can deselect one or more PAD files or you can click Deselect All, and then select the PAD files to be imported.

All selected PAD files are imported into the current Diagram Trace Utility debug session. You can view the PAD files in the PAD Code View editor.

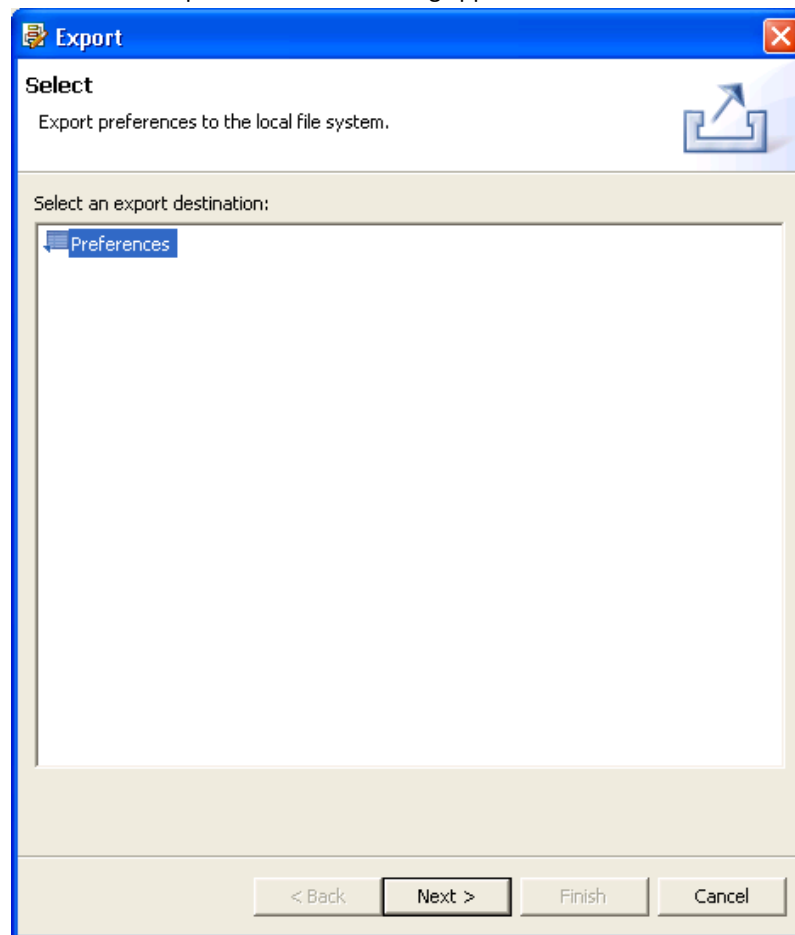
Export Preferences to a File

The Export Preferences feature provides a wizard that lets you export preference settings defined in the current Diagram Trace Utility debug session to a file on the local system. Use this feature to save customized preference settings for use in future debug sessions. You can use the Import Preferences feature to import the preferences from the saved export file.

Follow these steps:

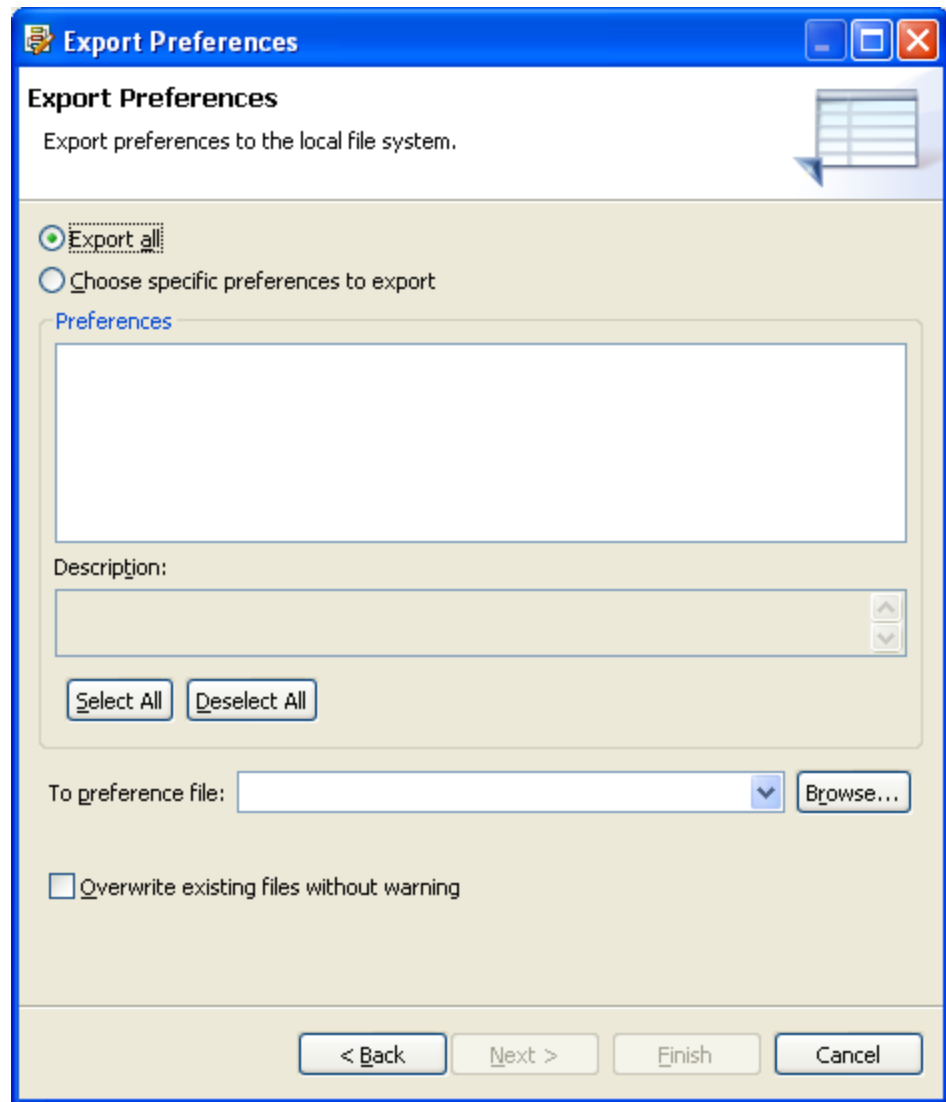
1. Select File, Export from the main menu.

The Select an Export Destination dialog appears.



2. Select Preferences and click Next.

The Export Preferences selection dialog appears.



3. Select Export all or select Choose specific preferences to export and select the preferences to export from the selection list.

Note: The Diagram Trace Utility does not support the Choose specific preferences to export option.

4. Click Browse and browse to the file (or create a file) in which to save the exported preferences.

Note: The file name of the exported file must have an .epf suffix.

5. Click Finish.

The preferences from the current Diagram Trace Utility debug session are saved in the file you specified.

Use Online Help

The Diagram Trace Utility provides extensive online documentation, context-sensitive help, and dynamic help. You can request online help during use of the Diagram Trace Utility anytime by clicking the Help menu and selecting the following options as needed.

Help Contents

Opens an external Help browser window and displays a top-level link of the Diagram Trace Utility online documentation, which you can expand to display subsets of the documentation.

Search

Activates the internal online Help view and displays the Search panel in which you can enter text you want to search for and click Go to display all topics in the online documentation in which the search term appears.

Dynamic Help

Activates the internal online Help view and displays all links to the online documentation that are related to the view that has the keyboard focus when you request help.

Online Support

Displays the SupportConnect web page, which provides information on CA online technical support.

About Diagram Trace

Displays the About Diagram Trace Utility dialog, which provides information on the software release number and copyright and contains a link to view the end-user license agreement.

More information:

[Navigate the Online Help System](#) (see page 149)

Context-Sensitive Help

If you encounter part of the Diagram Trace Utility interface that you do not understand, you can use context-sensitive help by clicking on a view tab and pressing F1 or selecting Dynamic Help from the Help menu.

F1 Support

You can access online Help for most (but not all) window controls by placing the focus on the dialog or control in question and pressing F1.

If the control in question is located in the Diagram Trace Utility main window, the Help view is activated automatically and a keyword search targeting text in the online documentation related to the control is performed. When the search is complete, links to topics related to the keyword search are displayed in the Help panel.

If the control in question is located on a dialog launched by the Diagram Trace Utility, an external Help window is activated. The same keyword search is performed and links to topics related to the keyword search are displayed in the Help panel.

Note: The F1 Help feature is supported in the Diagram Trace Utility views. This feature is not supported in the PAD Code View editor, some pop-up menus, window controls such as menu items or buttons, or toolbar icons.

Dynamic Help

When the internal Help view is active in the Diagram Trace Utility, as you move the focus from view to view, the help data displayed is dynamically updated to reflect the view that is in focus.

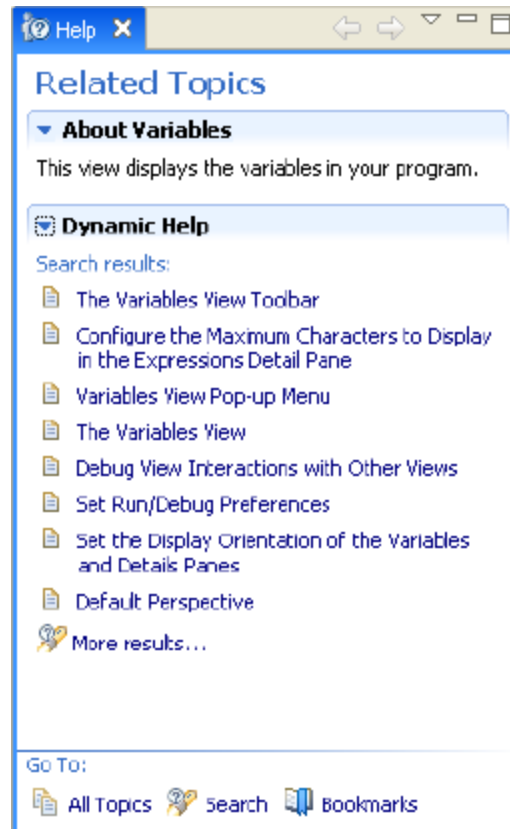
Note: The Dynamic Help feature is supported only for the views in the Diagram Trace Utility. This feature is not supported in the PAD Code View editor, pop-up menus, window controls such as menu items or buttons, or toolbar icons.

The internal Dynamic Help view contains the following four pages that present help topics in different ways:

- [Related Topics Page](#) (see page 79)
- [All Topics Page](#) (see page 80)
- [Search Page](#) (see page 81)
- [Bookmarks Page](#) (see page 82)

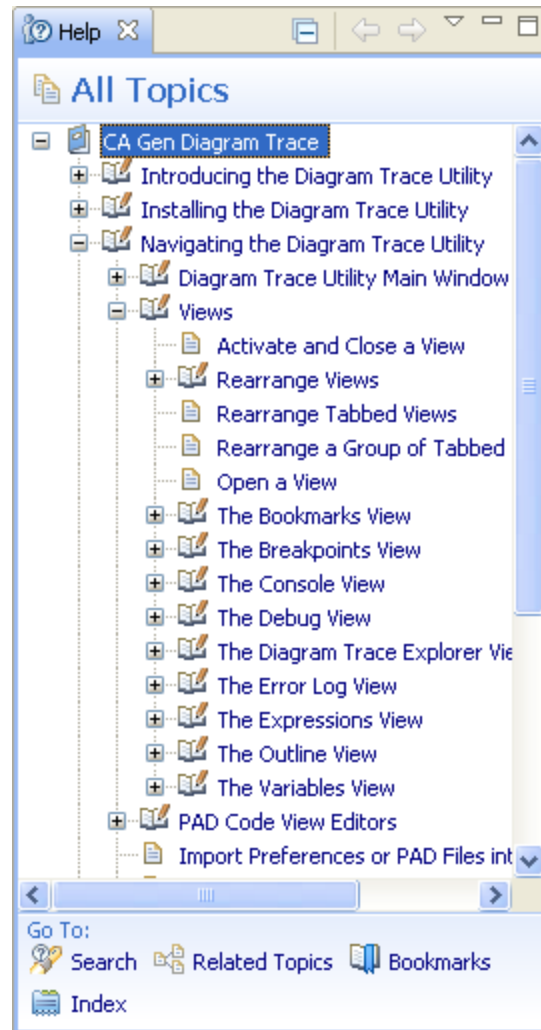
Related Topics Page

The Related Topics page shows description and links to topics in the online documentation related to the current context as shown in the following illustration. You can click on More Results to view more links.



All Topics Page

The All Topics page, shown in the following illustration, displays the Table of Contents for the online Diagram Trace Utility documentation arranged in a tree hierarchy that you can expand or collapse as needed.



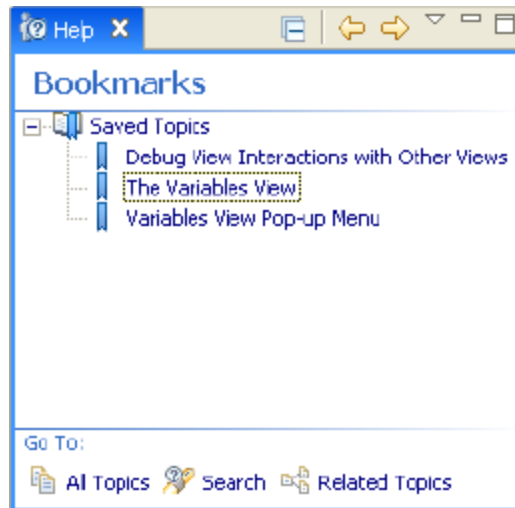
Search Page

The Search page, shown in the following illustration, lets you enter search text. When you click Go, the Diagram Trace Utility performs the search and displays links in the online documentation to the search hits along with a summary of topic contents.



Bookmarks Page

The Bookmarks page shown in the following illustration displays online Help topics that you mark as personal bookmarks.



Chapter 4: Preparing to Debug

This chapter describes how to configure a generated application and how to start, stop, and configure the Diagram Trace Utility to ensure successful connections between the generated application and the Diagram Trace Utility so that you can successfully start a debug session.

This section contains the following topics:

[Configure the Generated Application](#) (see page 83)

[Start the Diagram Trace Utility](#) (see page 84)

[Exit the Diagram Trace Utility](#) (see page 84)

[Start a Debug Session](#) (see page 85)

[Stop a Debug Session](#) (see page 85)

[Configure the Diagram Trace Utility](#) (see page 85)

Configure the Generated Application

You can use the Diagram Trace Utility to debug C, Java, or C# applications generated by CA Gen. These applications can be block mode applications, client applications, or server applications. You can also use the Diagram Trace utility to debug execution through the CA Gen Toolset Testing Tool. Specific configuration procedures and steps required to start a particular application depend on the application type and target runtime being used.

Note: You must start the Diagram Trace Utility before starting the target application to ensure a successful connection between the Diagram Trace Utility and the target application.

For more information on configuring your application, see the CA Gen documentation for the appropriate target runtime listed in the following table:

Language Runtime	Documentation
Block Mode applications	<i>Windows Implementation Toolset User Guide, UNIX and Linux Implementation Toolset User Guide, or NonStop Implementation Toolset User Guide</i>
C	<i>Transaction Enabler User Guide, Tuxedo User Guide, Distributed Processing - WebSphere MQ User Guide, or NonStop Implementation Toolset User Guide</i>
Java	<i>Web Generation User Guide or Distributed Processing - Enterprise JavaBean User Guide</i>

Language Runtime	Documentation
C#	<i>ASP.NET User Guide</i> or <i>Distributed Processing - .NET Server User Guide</i>
CA Gen Testing Tool	<i>Rapid Test Tool Help</i>

Start the Diagram Trace Utility

Start the Diagram Trace Utility before you start your generated application.

To start the Diagram Trace Utility, Click Start, All Programs, CA, Gen<xx>, Diagram Trace Utility. The Diagram Trace window opens.

Note: xx refers to the current release of CA Gen.

The Diagram Trace Utility is pre-configured to start automatically when it is initialized. You can change this behavior through the Prestart option on the Diagram Trace Utility Preferences dialog.

Note: If you have stopped the Diagram Trace Utility, you can restart it by clicking the Start button on the main window toolbar.

Exit the Diagram Trace Utility

The Diagram Trace Utility automatically saves the window layout or perspective each time you exit the utility. If you change the default perspective during a debug session, the changes are automatically saved as the current perspective, including all open views, PAD Code View editors, and the current window layout. The next time you start the Diagram Trace Utility, the current perspective is displayed. The main window appears exactly as it was the last time you closed it.

You can take any of the following actions to exit the Diagram Trace Utility:

- Select Exit from the File menu on the main window.
- Right-click the main window title bar and select Close from the system pop-up menu.
- Click the X icon on the right side of the main window title bar.
- Press Alt+F4 when the Diagram Trace Utility has the current application focus.

Start a Debug Session

When you start the Diagram Trace Utility, it is pre-configured to start a debug session automatically. You can change this behavior through the **Prestart** option on the Diagram Trace Utility Preferences dialog.

Note: If you have stopped the debug session, you can click the **Start** button on the toolbar of the main window to restart the session.

Stop a Debug Session

You can stop the current debug session anytime by clicking the **Stop** button on the toolbar of the main window.

Configure the Diagram Trace Utility

The Diagram Trace Utility uses the TCP/IP communications protocol. Before you use the Diagram Trace Utility, verify that TCP/IP is installed and configured properly on the target application machine and the machine hosting the Diagram Trace Utility.

The Diagram Trace default configuration specifies port number 4567 to communicate with target applications. You can change this default port or add other ports using the Diagram Trace Preferences dialog. You can also configure the Diagram Trace Utility settings using the Preferences dialog.

More information:

[Set General Startup Preferences for Diagram Trace Utility](#) (see page 108)

Chapter 5: Running a Debug Session

The Diagram Trace Utility can communicate simultaneously with one or more generated applications locally or across a network using TCP/IP. The Diagram Trace Utility listens on its configured port number for connections from applications that have been started with the Trace mode enabled. Applications connect to the Diagram Trace Utility using the host and port number you configure as trace enabling options. Multiple applications or multiple instances of the same application can connect concurrently to the same Diagram Trace Utility port or to separately configured ports. Depending on the type of application being traced, the Diagram Trace Utility may create additional connections. The Diagram Trace Utility typically maintains these connections and closes them as needed.

You can manually close any of these connections. However, if you manually close a connection, the application continues execution without tracing. Depending on the type of generated application being traced, tracing may be resumed. In general, Web applications resume tracing as soon as a new browser Request is initiated.

Important! C-based applications do not resume tracing. You must restart C-based applications.

After an application connects to the Diagram Trace Utility, CA Gen PAD language statements sent from the application display in the PAD Code view of the Diagram Trace main window. The Diagram Trace Utility does not need access to generated application source code files. However, you must generate the application to be debugged with the Trace option enabled.

Note: You must start Diagram Trace before starting the target application to ensure a successful connection between the Diagram Trace Utility and the target application.

This section contains the following topics:

[Debug Session Overview](#) (see page 88)

[Configure and Start a Debug Session](#) (see page 89)

[Start the Generated Application](#) (see page 89)

[How a Traced Program Executes](#) (see page 94)

[Use Breakpoints to Control Program Execution](#) (see page 94)

[Manage Bookmarks](#) (see page 104)

[Exit the Debug Session](#) (see page 106)

Debug Session Overview

To initiate and run a basic Diagram Trace debugging session, perform the following tasks:

- [Configure and start one or more debug configurations](#) (see page 89)
- [Start the generated application](#) (see page 89)
- [Add a line breakpoint](#) (see page 96)
- [Enable a breakpoint](#) (see page 96)
- [Disable a breakpoint](#) (see page 96)
- [Remove a breakpoint in the Breakpoints View](#) (see page 97)
- [Remove a breakpoint from the vertical ruler of a PAD Code View Editor](#) (see page 97)
- [Set hit counts and conditional breakpoints](#) (see page 99)
- [Hit Counts](#) (see page 98)
- [Set a Hit Count](#) (see page 99)
- [Add a method breakpoint](#) (see page 101)
- [Enable a method breakpoint](#) (see page 102)
- [Disable a method breakpoint](#) (see page 102)
- [Remove a method breakpoint](#) (see page 103)
- [Set a Suspend on initial entry breakpoint](#) (see page 103)
- [Add a bookmark](#) (see page 105)
- [Access an Action Diagram with a bookmark](#) (see page 105)
- [Remove a bookmark](#) (see page 106)
- [Exit the debug session](#) (see page 106)

Configure and Start a Debug Session

You must start a debug session in the Diagram Trace Utility before you can trace an application.

Follow these steps:

1. Click Start, All Programs, CA, Gen xx, and select Diagram Trace Utility.

The Diagram Trace Utility main window opens.

Note: If you have not changed the default configuration, the Diagram Trace Utility automatically starts a debug session when you start the software. If you have changed the default configuration, you must start the debug session manually.

If you are using the default port number (4567) and you need to start the debug session manually, move to Step 4. To add or change a port number or change the pre-start Diagram Trace Utility preference, perform the tasks in Step 2 and Step 3.

2. Click Windows, Preferences.

The Preferences dialog opens.

3. Change the value of the Listen on Ports field and click OK.

Note: You can type multiple port numbers, separated by spaces.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

4. Click Diagram Trace, Start if you have changed the default configuration and need to start the utility manually.

A listener thread starts for each port number configured and each configured thread is listed in the Debug view.

Note: You can use the Windows `netstat -a` command to display a list of TCP/IP ports currently in use on the machine hosting the Diagram Trace Utility. Reviewing this list of active ports helps you resolve conflicts with port numbers that are already in use.

Start the Generated Application

The procedure used to start a particular generated application depends on the application's targeted language runtime (C, Java, or C#).

More information:

[Prerequisites](#) (see page 90)

[Prepare the Application for Trace](#) (see page 90)

[Start the Generated Application on Windows](#) (see page 91)

[Start the Generated Application on UNIX and Linux](#) (see page 92)

[Start the Generated Application on NonStop](#) (see page 93)

[Verify the Diagram Trace Utility Connection](#) (see page 93)

Prerequisites

Before you start the application, you must have generated the application in the CA Gen Toolset or Encyclopedia with Action Diagram tracing enabled.

Java and C# CA Gen applications are installed using an application deployment .msi file that the CA Gen Build Tool creates. When the application .msi files are created through the Build Tool, you must enable tracing and specify the Diagram Trace Utility server's address and port number.

Important! The port number specified by the target application must match the port number configured in the Diagram Trace Utility.

Prepare the Application for Trace

Before you trace an application on a Windows, UNIX, Linux, or a NonStop system, you must build certain components using the Build Tool. The following table details the specific components that you must build on a Windows, UNIX, Linux, or a NonStop system.

Operating System	Components
Windows	The application database
	RI trigger logic
	Any applicable operations libraries
	All load modules
UNIX and Linux	The application database
	RI trigger logic
	Any applicable operations libraries
	All load modules

Operating System	Components
NonStop	The application database RI trigger logic All load modules

Start the Generated Application on Windows

You can trace all the generated C applications using the Diagram Trace Utility on Windows Systems. Each application requires transaction codes to map to the correct procedure steps and action blocks. The following table lists the C applications that you can invoke and trace:

Language Runtime	Invocation Method	How to Obtain Trancode
GUI Windowed	Command Line	As a parameter.
	Windows Explorer	As the first trancode stored within the executable
Distributed Processing Client (GUI Client)	Command Line	As a parameter
	Windows Explorer	As the first trancode stored within the executable
Distributed Processing Server	AEFAD or WebSphere MQ	From the AEENV file
Block Mode Command line	Command Line	As a parameter when invoked and then from the AEENV file during execution
Block Mode Interactive	AEFN	From the AEENV file

Start the Generated Application on UNIX and Linux

You can remotely trace all the generated C applications on UNIX and Linux Systems remotely using the Diagram Trace Utility running on Windows. For each application, set the AEPATH environment variable when invoking the application through AEF. The following table lists the C applications that you can invoke and trace:

Language Runtime	Invocation Method	AEPATH
Distributed Processing Server	AEFAD, WebSphere, MQ or Tuxedo	<p>AEFAD—Set before you start AEFAD</p> <p>Tuxedo—Set before issuing tmboot</p> <p>WebSphere MQ—Set before starting individual load module</p> <p>Create a .ksh file to launch the AEFAD and AEFUF before you execute the C Servers.</p> <p>Sample .ksh file:</p> <pre>export AEPATH `pwd` ":\$"IEFH aefad -t 15 aefuf -t 15</pre> <p>Execute the .ksh file from the model directory to set the current directory in AEPATH.</p> <p>Create a similar .ksh file for launching Tuxedo and WebSphere MQ servers.</p>
Block Mode Commandline	Command Line	Set before invoking an application
Block Mode Interactive	AEF	Set before invoking AEF

Start the Generated Application on NonStop

All generated C applications on NonStop Systems can be traced remotely using the Diagram Trace Utility running on Windows. Each application requires transaction codes in order to map to the correct procedure steps and action blocks. The following table lists the C applications that can be traced, and how they are invoked:

Language Runtime	Invocation Method	How to Obtain Trancode
Distributed Processing Server	Pathway through RSC/MP	From transaction mapping table
Block Mode Command line No Display	Pathway through non-TCP Requester	From transaction mapping table
Block Mode Interactive	Pathway through AEF	From transaction mapping table

Verify the Diagram Trace Utility Connection

When you start any type of application (block mode, C, Java, or C#), it automatically attempts to connect to the Diagram Trace Utility server and one of the following events occurs:

- If the connection is successful, you can view the application's PAD code in the PAD Code view editor, which gets updated when an action is performed in the application that causes a ProcedureStep to execute.
- If the connection is unsuccessful, the application connection to the Diagram Trace Utility times out and application execution proceeds as if trace was not enabled.

For more information about starting applications, see the appropriate documentation for your targeted environment listed in the following table:

Language Runtime	Documentation
Block Mode applications	<i>Windows Implementation Toolset User Guide, UNIX and Linux Implementation Toolset User Guide, or NonStop Implementation Toolset User Guide</i>
C	<i>Transaction Enabler User Guide, Tuxedo User Guide, Distributed Processing - WebSphere MQ User Guide, or NonStop Implementation Toolset User Guide</i>
Java	<i>Web Generation User Guide or Distributed Processing - Enterprise JavaBean User Guide</i>

Language Runtime	Documentation
C#	<i>ASP.NET User Guide</i> or <i>Distributed Processing - .NET Server User Guide</i>
CA Gen Testing Tool	<i>Rapid Test Tool Help</i>

How a Traced Program Executes

When you step through the execution of a traced program, the following events occur:

1. The application started in trace mode successfully attaches itself to the Diagram Trace Utility. A subset of the Action Diagram being traced displays in the PAD Code View editor window.
2. The trace execution stops upon entering the initial Action Diagram before any code is executed. This is the default behavior.

Note: You can change the default behavior by clearing the option *Suspend on initial entry* in the Diagram Trace Preferences dialog or clicking the "Suspend on entry" toggle button on the debug view toolbar.

Use Breakpoints to Control Program Execution

Breakpoints suspend program execution where the breakpoint is set. Breakpoints are displayed in the vertical ruler in the PAD Code View editor and in the Breakpoints view. Blue circles identify enabled breakpoints and white circles identify disabled breakpoints.

You can enable and disable breakpoints in the Breakpoints view or in the PAD Code View editor window. When a breakpoint is enabled, it suspends program execution whenever the breakpoint is reached. When a breakpoint is disabled, execution is not suspended when the breakpoint is reached.

When the program execution is suspended as a result of encountering a breakpoint, you can use the Debug view step controls to step through the execution of the program statement-by-statement. If the Diagram Trace Utility encounters a breakpoint while performing a step operation, it suspends program execution at the breakpoint and ends the step operation.

The Diagram Trace Utility provides the following line breakpoints to help you manage the step execution in the application program being traced.

Step Over

Resumes execution, stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Step Into

Resumes execution, stopping at the next statement either in the same method, a called method, or (if you are at the end of a method) in the method from which the current method was called.

Step Return

Resumes execution, stopping at the next statement in the method from which the current method was called.

Step with Filters

Applies a set of step filters defined on the Diagram Trace Step Filtering Preferences page. Use the Use Step Filters button in the Debug view toolbar to turn Step with Filters on and off.

When the action is turned on, each of the step actions (over, into, return) applies the set of step filters defined on the Diagram Trace Step Filtering Preferences page, which you can access from the main menu bar by selecting Window, Preferences, Diagram Trace, Step Filtering. When a step action is invoked, stepping continues until the execution reaches an unfiltered location or encounters a breakpoint.

Run to Line

Suspends execution at a line without setting a breakpoint. This feature requires a source line to be selected before you select Run to Line. Execution of a suspended program resumes until the selected line is executed.

It is possible that the line may never be encountered and that the program will not suspend. Breakpoints and exceptions can cause the program to suspend before reaching the specified line.

Skip Over

Resumes the application (without executing the current statement), stopping at the next statement either in the same method or (if you are at the end of a method) in the method from which the current method was called.

Skip to Line

Continues execution from the current statement up to the line containing the cursor and skips this line instead of executing it. You can also use this feature to skip execution of a sequence of statements.

More information:

[Set Diagram Trace Step Filtering Preferences](#) (see page 115)

Add a Line Breakpoint

You can set a line breakpoint on an executable statement associated with a line of code in an Action Diagram. A line breakpoint, when enabled, suspends program execution before a specified statement in the code is executed. The line where the breakpoint is set is highlighted in the PAD Code View editor. The current stack trace is displayed in the Debug view.

Perform one of the following steps:

- Place your cursor on the vertical ruler in the PAD Code View editor directly to the left of the line in which you want to add the breakpoint, and select **Toggle Line Breakpoint** from the Run menu.
- Double-click the mouse on the vertical ruler to the left of the line in which you want to add the breakpoint.
- Place your cursor over the vertical ruler, right-click, and select **Toggle Breakpoint** from the pop-up menu.

Note: When you add a breakpoint, it is enabled by default.

Enable a Breakpoint

You can enable a breakpoint in the Breakpoints view as necessary.

To enable a breakpoint, select the breakpoint, right-click, and select **Enable Breakpoint** from the pop-up menu. The icon associated with the breakpoint changes from a non-filled circle to a filled circle.

While the breakpoint is enabled, program execution suspends before that statement is executed. The line where the breakpoint is set is highlighted in the Debug editor. The current stack trace is displayed in the Debug view.

Note: When you add a breakpoint, it is enabled by default.

Disable a Breakpoint

You can disable a breakpoint in the Breakpoints view as necessary.

To disable a breakpoint, select the breakpoint, right-click, and select **Disable Breakpoint** from the pop-up menu. The breakpoint icon changes from a filled circle to a non-filled circle.

When a line breakpoint is disabled, the program execution is not suspended where the breakpoint is set. The execution of the program continues as if the breakpoint was never set.

Remove a Breakpoint in the Breakpoints View

You can remove one or more breakpoints in the Breakpoints view when you no longer need them.

To remove breakpoints in the Breakpoints view, select one or more breakpoints, right-click, and select **Remove** from the pop-up menu.

To remove all breakpoints in the Breakpoints view, select **Remove All** from the pop-up menu.

Remove a Breakpoint from the Vertical Ruler of a PAD Code View Editor

You can remove a breakpoint that is either enabled or disabled.

Follow these steps:

- Select the location on the vertical ruler in the PAD Code View editor area containing the breakpoint you want to remove and select **Toggle Line Breakpoint** from the Run menu.
- Double-click on the breakpoint icon on the vertical ruler of the PAD Code View editor.
- Right-click on the breakpoint icon on the vertical ruler and select **Toggle Breakpoint** from the pop-up menu.

Note: If you find yourself frequently removing and adding a breakpoint in the same place, consider disabling and enabling the breakpoint in the Breakpoints view.

Set the Suspend on Initial Entry Preference

You can set the suspend on initial entry preference value.

Follow these steps:

1. Select Window, Preferences...

The Preferences dialog appears.

Note: You can also click Suspend on entry toggle button in the Debug View toolbar to set the suspend on initial entry preference value.

2. Select Diagram Trace in tree view.

Select Suspend on initial entry and click OK.

3. The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Note: If this preference has been disabled, you can enable it again through the Preferences dialog or the debug view toolbar toggle button Suspend on entry.

Set Hit Counts and Conditional Breakpoints

The following sections describe setting the hit count and conditional breakpoint features.

Hit Counts

When you apply a hit count to a breakpoint, the breakpoint suspends program execution when the specified Hit Count condition is satisfied. You can apply a hit count to line breakpoints, method breakpoints, and watchpoints.

Following are the valid Hit Count conditions:

Break always

Stops program execution each time the Diagram Trace Utility encounters the breakpoint.

Break when the hit count is equal to <value>

Stops program execution when the Diagram Trace Utility encounters the breakpoint the number of times you specified in the Hit Count field.

Break when the hit count is a multiple of <value>

Stops program execution when the Diagram Trace Utility encounters a breakpoint a multiple of the number of times you specified in the Hit Count field.

Break when the hit count is greater than or equal to <value>

Stops program execution when the Diagram Trace Utility encounters the breakpoint a number of times equal to or greater than the number of times you specified in the Hit Count field.

The Current Hit Count control displays the current Hit Count value. To reset this value, click Reset Hit Count.

Set a Hit Count

You can set a hit count on a breakpoint from the Breakpoints view pop-up menu or from the pop-up menu of the vertical ruler in the PAD Code View editor.

Follow these steps:

1. Select the breakpoint to which a hit count is to be added. Right-click and select one of the following pop-up menu options depending on the view you are in:

- Properties (from the Breakpoints view)
- Breakpoint Properties (on a selected breakpoint in the Vertical ruler bar of the PAD Code View editor)

The Breakpoint Properties dialog appears.

2. Do the following:
 - a. Select hit count in the Hit Count combo box.
 - b. Enter a number in the Hit Value field.

Note: This value specifies the number of times you want to hit the breakpoint before suspending execution.

3. Click OK.

The Diagram Trace Utility sets the hit count for the breakpoint.

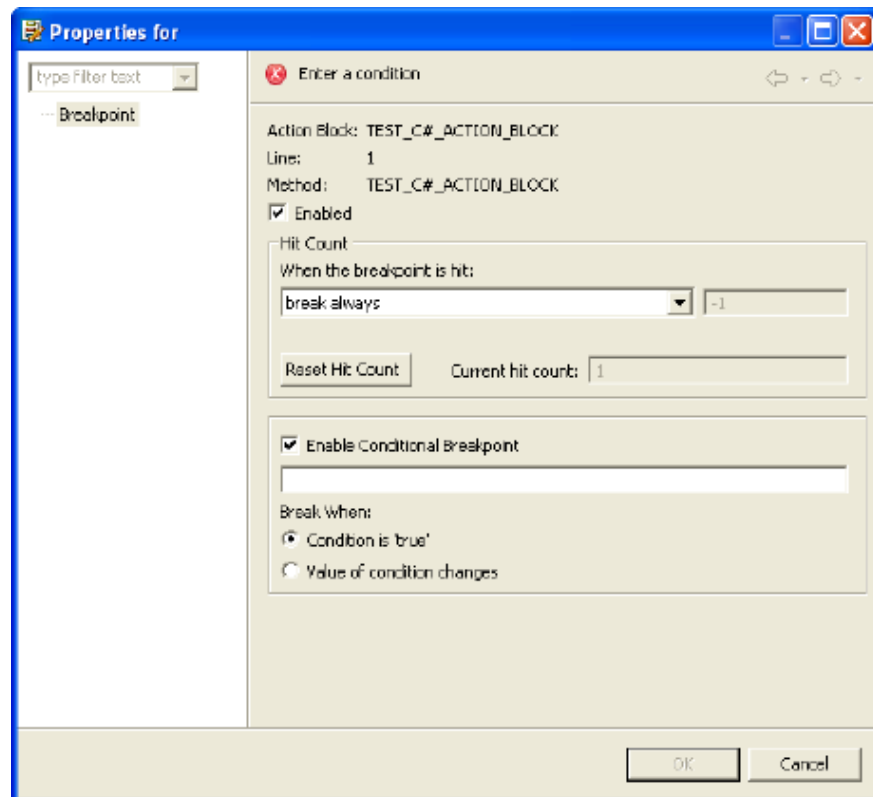
Set a Condition on a Line Breakpoint

You can set a condition on a breakpoint so that the breakpoint suspends program execution only when the condition is satisfied.

Follow these steps:

1. Do one of the following steps:
 - In the Breakpoints view, right-click the breakpoint on which you want to set a condition, and select Properties.
 - In the vertical ruler of the PAD Code View editor, right-click the breakpoint on which you want to set a condition, and select Breakpoint Properties from the pop-up menu.

The Breakpoint Properties dialog appears.



2. Select Enable Conditional Breakpoint.
The text field for Enable Conditional Breakpoint is enabled.
3. Enter a boolean expression in the Enable Conditional Breakpoint field.

4. Click one of the following Break When option buttons:

Condition is true

The breakpoint suspends program execution before the line of code containing the breakpoint is executed each time the condition evaluates to true.

Value of condition changes

The breakpoint suspends program execution only when the result of the condition changes.

5. Click OK.

The Diagram Trace Utility saves the changes and closes the dialog.

More information:

[Expressions](#) (see page 14)

Supported Conditions Statements on a Line Breakpoint

Supported condition statements include any valid expressions.

More information:

[Expressions](#) (see page 14)

Add a Method Breakpoint

You can add a method breakpoint to stop application execution when entering an Action Diagram or Event Action. After you have added one or more method breakpoints, you can enable, disable, or remove them just like you can enable, disable, and remove line breakpoints.

Follow these steps:

Perform one of the following steps:

- Select the Action Diagram in the Outline view, and select Toggle Method Breakpoint from the Run menu.
- Select the Action Diagram in the Outline view, right-click your mouse, and select Toggle method Breakpoint from the Outline view pop-up menu.
- Select the Action Diagram or Event Action in the PAD Code View editor, and select Toggle Method Breakpoint from the Run menu.

A breakpoint appears in the Breakpoints view. A breakpoint marker also appears in the vertical ruler at the first source line in the file's PAD Code View editor for the selected Action Diagram or Event Action.

Note: When you add a method breakpoint, it is enabled by default.

While the breakpoint is enabled, the Diagram Trace Utility suspends program execution when it enters an Action Diagram or Event Action, before any line in the method is executed.

You can also modify the breakpoint properties to suspend the execution upon exiting a method (that is, before leaving an Action Diagram or an Event Action).

Enable a Method Breakpoint

You can enable a disabled method breakpoint by selecting the Action Diagram in the Breakpoints view or in the vertical ruler of the PAD Code View editor.

Follow these steps:

- Select the Action Diagram in the Breakpoints view and check the check box of the disabled breakpoint to enable it.
- Select the Action Diagram in the vertical ruler of the PAD Code View editor, right-click and select Enable Toggle Method Breakpoint from the Outline view pop-up menu.

The Diagram Trace Utility suspends program execution when it enters an Action Diagram or Event Action, before any line in the method is executed.

Disable a Method Breakpoint

You can disable a defined, enabled method breakpoint in the Breakpoints view or in the PAD Code View editor's vertical ruler.

Note: The breakpoint icon image is a circle. The circle is unfilled if the breakpoint is disabled or is filled if the breakpoint is enabled.

To disable a method breakpoint in the Breakpoints view, select the method breakpoint, right-click, and select Disable from the pop-up menu.

To disable a method breakpoint in the vertical ruler of a PAD Code View editor

Follow these steps:

1. Right-click the breakpoint icon in the vertical ruler.

The Breakpoint's pop-up menu appears.

2. Select Disable Breakpoint.

The breakpoint icon image changes from a filled circle to an unfilled circle.

Remove a Method Breakpoint

You can remove one or more selected method breakpoints from the Breakpoints view or from the vertical ruler in a PAD Code View editor in the same way that you remove a line breakpoint.

You can also remove a selected method breakpoint in the Outline view.

Follow these steps:

1. Select Show View, Outline from the Windows menu on the main menu bar.

The Outline view appears.

2. Select the Action Diagram from which you want to remove a method breakpoint, and perform one of the following steps:

- Select Toggle Method Breakpoint from the Run menu on the main menu bar.
- Right-click and select Toggle Method Breakpoint from the Outline view Pop-up menu.

The Diagram Trace Utility removes the method breakpoint from the selected Action Diagram in the Breakpoints view.

Set a Suspend on Initial Entry Preference

You can set a Diagram Trace preference to stop program execution at the entry of each Action Diagram and Event Action encountered. This preference acts as if a breakpoint is set on the first line of each Action Diagram and Event Action in the application being traced. However, no breakpoint icon is visible on the vertical ruler. This preference is enabled by default.

Note: If this preference has been disabled, you can enable it again through the Preferences dialog.

Follow these steps:

1. Select Preferences from the Window menu on the Diagram Trace main Window.
The Preferences dialog appears.
2. Click the Diagram Trace tree view item on the left, select Suspend on initial entry, and click OK.
The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Manage Bookmarks

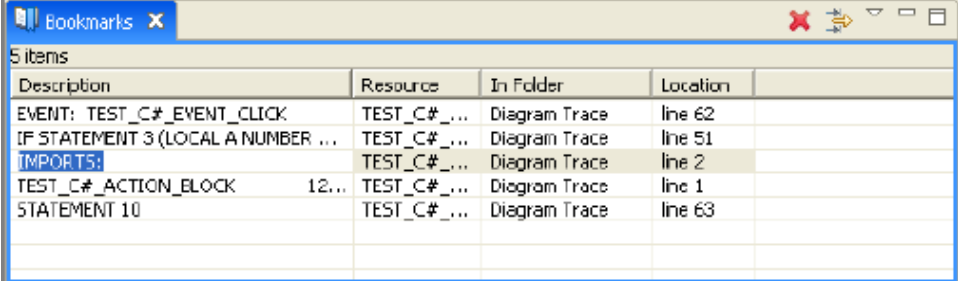
Bookmarks help you to navigate quickly in the source code. The Diagram Trace Utility allows you to bookmark individual locations in a generated Action Diagram. You can add a bookmark view to the current perspective and add and delete bookmarks as needed.

Add the Bookmarks View to the Current Perspective

The Bookmarks view displays all bookmarks you have created.

To add the Bookmarks view to the current perspective, select Show View, Bookmarks from the Window menu.

The Bookmarks view displays the bookmarks you created.



Description	Resource	In Folder	Location
EVENT: TEST_C#_EVENT_CLICK	TEST_C#_...	Diagram Trace	line 62
IF STATEMENT 3 (LOCAL A NUMBER ...	TEST_C#_...	Diagram Trace	line 51
IMPORTS:	TEST_C#_...	Diagram Trace	line 2
TEST_C#_ACTION_BLOCK 12...	TEST_C#_...	Diagram Trace	line 1
STATEMENT 10	TEST_C#_...	Diagram Trace	line 63

Following is the description of the different columns in the Bookmarks view:

- The Description column contains a description of the bookmark. To change the description, right-click, and select Properties from the pop-up menu.
- The Resource column lists the .pad file name that the Diagram Trace Utility uses to store a cached version of the Action Diagram being traced.
- The In Folder column specifies the folder in which the resource resides.
- The Location column indicates the line number of the bookmark in its Action Diagram.

Add a Bookmark

You can add bookmarks to your Action Diagram to help you navigate quickly in the PAD statements.

Follow these steps:

1. Select Show View, Bookmarks from the Window menu.

The Bookmarks view appears in the Diagram Trace Utility main window.

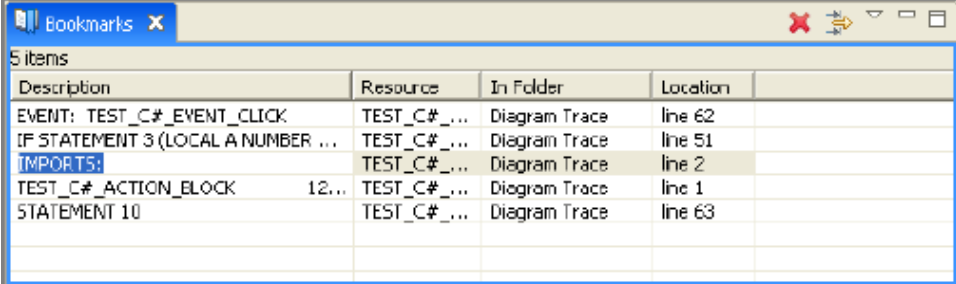
2. Place your cursor over the PAD Code View editor's vertical ruler bar next to a line in the file. Right-click and select Add Bookmark from the pop-up menu.

The Add Bookmark dialog appears.

3. Enter a description for the new bookmark and click OK.

The new bookmark is added to the Bookmarks view.

Example: The following illustration shows a sample bookmarks that has been added to the Bookmarks view.



The screenshot shows a window titled 'Bookmarks' with a table of bookmarked items. The table has five columns: Description, Resource, In Folder, Location, and an empty column. The rows list various code elements and their locations in the diagram trace.

Description	Resource	In Folder	Location	
EVENT: TEST_C#_EVENT_CLICK	TEST_C#_...	Diagram Trace	line 62	
IF STATEMENT 3 (LOCAL A NUMBER ...	TEST_C#_...	Diagram Trace	line 51	
IMPORTS:	TEST_C#_...	Diagram Trace	line 2	
TEST_C#_ACTION_BLOCK 12...	TEST_C#_...	Diagram Trace	line 1	
STATEMENT 10	TEST_C#_...	Diagram Trace	line 63	

Access an Action Diagram with a Bookmark

After you create one or more bookmarks, you can access the bookmarked Action Diagrams using the bookmarks in the Bookmarks view.

To access the Action Diagram associated with a bookmark in the Bookmarks view, double-click the bookmark you created. If not already open, a PAD Code View editor opens and displays the Action Diagram with which the bookmark is associated. The line associated with the bookmark is highlighted.

Remove a Bookmark

After you create one or more bookmarks, you can remove bookmarks as necessary.

Follow these steps:

Select a bookmark and perform one of the following steps:

- Right-click your mouse, and select **Delete** from the pop-up menu.
- Click the **Delete** icon on the Bookmarks view toolbar.

The bookmark is removed from the Bookmarks view.

Exit the Debug Session

After you have finished debugging your application, you can disconnect the Listener process, terminate the Diagram Trace Utility server, or exit the Diagram Trace Utility by selecting **File, Exit** from the main menu bar.

You can take any of the following actions to exit the Diagram Trace Utility:

- Select **Exit** from the **File** menu on the main window.
- Right-click the main window title Bar and select **Close** from the system pop-up menu.
- Click the **X** icon on the right side of the main window title bar.
- Press **Alt+F4** when the Diagram Trace Utility has the current application focus.

Chapter 6: Customizing the Diagram Trace Utility

This chapter describes how to customize the appearance and behavior of the Diagram Trace Utility.

This section contains the following topics:

[Change the Diagram Trace Utility Main Window Layout](#) (see page 107)

[Set Diagram Trace Preferences](#) (see page 107)

[Set General Preferences](#) (see page 117)

Change the Diagram Trace Utility Main Window Layout

When you initiate a debug session, the Diagram Trace Utility displays its default perspective (window layout) in the Diagram Trace Utility main window.

Follow these steps:

Use one or more of the following methods:

- Drag one or more views to different positions in the Diagram Trace Utility main window
- Drag a PAD Code View editor to a position to enable simultaneous viewing above, below, or beside another editor
- Resize views and PAD Code View editors by dragging the sashes that separate them
- Open or close a view or a PAD Code View editor

These actions create a new perspective that is saved automatically when you exit the Diagram Trace Utility main window. The next time you start the Diagram Trace Utility, the previously saved perspective is displayed.

Set Diagram Trace Preferences

The Diagram Trace Preferences dialog lets you set general startup preferences for the Diagram Trace Utility, general Folding and Syntax Coloring preferences for the PAD Code View editor, and general preferences for Step Filtering.

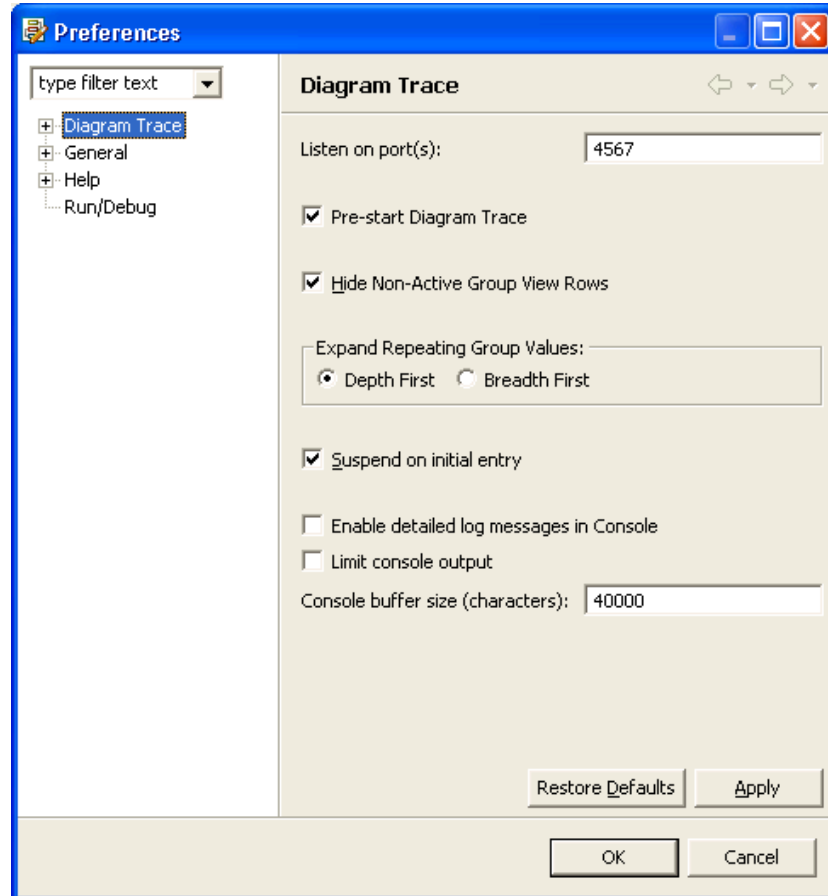
Set General Startup Preferences for Diagram Trace Utility

You can set the general startup preferences for the Diagram Trace Utility through the Diagram Trace Preferences page.

Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.



2. Change the default settings for the following fields to suit your needs:

Listen on Port(s)

Lets you change the default port number. You can add one or more port numbers separated by commas, spaces, or tabs if you want to configure multiple ports for tracing multiple applications concurrently. Multiple port numbers must contain only numbers 0 through 9.

Note: Concurrent debug sessions can use the same port number or separate port numbers. There is no limit to the number of concurrent debug sessions that can be active at one time. However, system resources will determine the practical limit.

Default: 4567

Limits: Numeric value

Prestart Diagram Trace

Starts a debug session automatically when the Diagram Trace Utility is started.

Default: On

Hide Non-Active Group View Rows

Hides the display of non-active group view rows.

Default: On

Expand Repeating Group values

Specifies how Variables view expands repeating group values.

The options are:

- Depth First—Drills down into individual instances as records.
- Breadth First—Drills down through occurrences until it reaches the bottom attributes and displays them as arrays of values instead of record fields.

Default: Depth First

Suspend on initial entry

Suspends the application execution whenever an Action Diagram is first entered.

Default: On

Enable detailed log messages in Console

Enables the display of detailed messages for standard input, output, and error messages logged in the Console view.

Default: Off

Limit console output

Limits the console buffer size to the value specified in the Console buffer size (characters) field.

Default: 40000

Limits: Integer between 1000 and 1000000

3. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog. The changes are used in all succeeding debug sessions unless you change the preferences again.

Note: You can restore the default configuration settings anytime on the Diagram Trace Preferences dialog. To restore the default settings, click Restore Defaults, click Apply, and then click OK.

Set Diagram Trace PAD Code View Editor Preferences

You can set the general preferences for the Diagram Trace Utility PAD Code View editor through the Diagram Trace Preferences page.

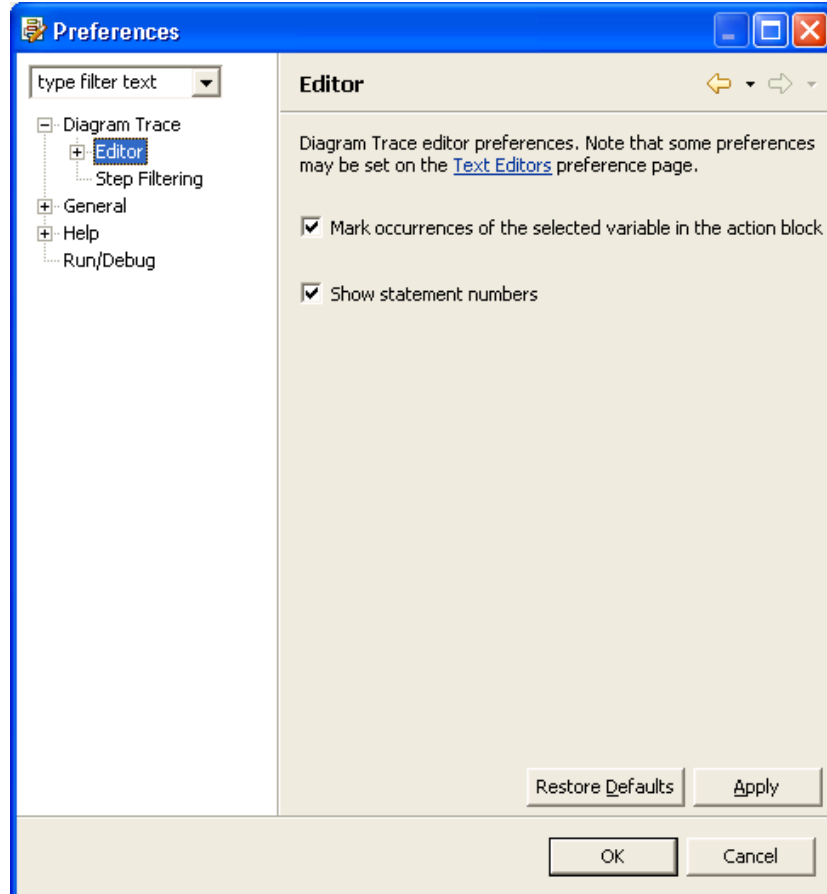
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to Diagram Trace, Editor.

The Editor Preferences page appears.



3. Change the default settings for the following fields to suit your needs:

Mark occurrences of the selected variable in the action block

Enables double-clicking on a Variableview in the PAD Code View editor to cause all occurrences of that variable to be marked and highlighted in the text area and appear as annotations in the overview ruler.

Default: On

Show statement numbers

Displays statement numbers for each line in the PAD Code View editor.

Default: On

4. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Set Diagram Trace Editor Folding Preferences

You can modify the folding behavior seen in the PAD Code View editor. Folding collapses the tree structure of specified elements in the PAD Code View editor. Unfolding expands the tree structure.

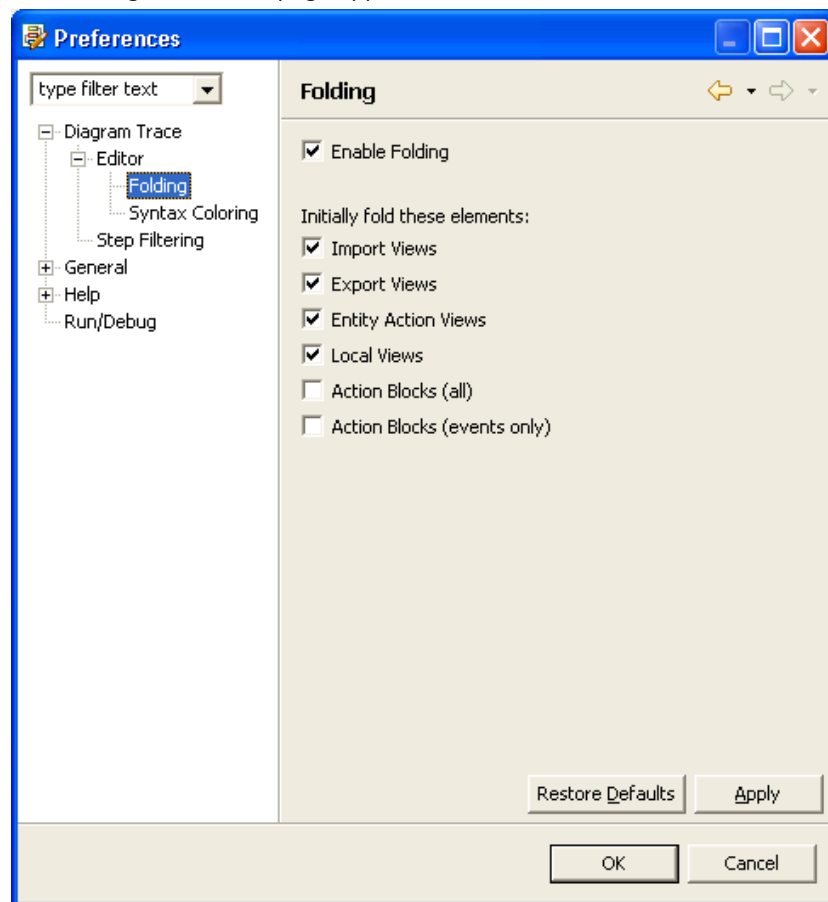
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to Diagram Trace, Editor, Folding.

The Folding Preferences page appears.



3. Change the default settings for the following fields to suit your needs:

Enable Folding

Enables folding in the PAD Code View editor.

Default: On

Initially fold these elements

Specifies the elements for initial folding.

Following elements are available:

- Import Views
- Export Views
- Entity Action Views
- Local Views
- Action Blocks (all)
- Action Blocks (events only)

Note: Initial folding does not affect PAD Code View editors that are currently open.

4. Click OK.

The system saves the changes and closes the Preferences dialog.

Set Editor Syntax Coloring Preferences

Syntax coloring affects the way text for different elements in the PAD Code View editor is displayed.

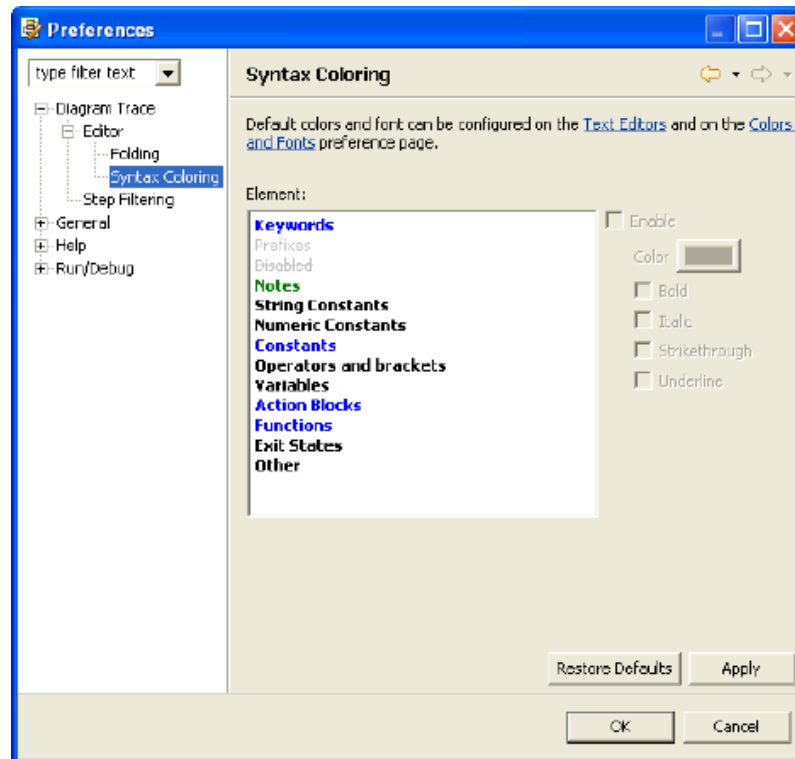
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to Diagram Trace, Editor, Syntax Coloring.

The Syntax Coloring Preferences page appears.



The Syntax Coloring Preferences page lets you view and change the default settings for the listed elements that appear in the PAD Code View editor. You can click each element to see if the Enable check box is checked (to enable syntax coloring) and view the color attribute setting for each element. The following syntax coloring default status, colors, and attributes are defined for the listed elements:

- Keywords—Enabled, Blue, Bold
- Prefixes—Enabled, Light Grey
- Disabled—Enabled, Light Grey, Strikethrough
- Notes—Enabled, Dark Green, Bold

- String Constants—Disabled, Black, Bold
 - Numeric Constants—Disabled, Black, Bold
 - Constants—Enabled, Blue, Bold
 - Operators and Brackets—Disabled, Black, Bold
 - Variables—Disabled, Black, Bold
 - Action Blocks—Enabled, Blue, Bold
 - Functions—Enabled, Blue, Bold
 - Exit States—Disabled, Black, Bold
 - Other—Enabled, Black Bold
3. (Optional) Select the element from the list whose color you want to change and click the Color button to the right of the element list.

A pop-up Color selection dialog displays from which you can select a basic color or custom color.
 4. (Optional) Select the element from the list whose attribute you want to change and select any of the following options:
 - Bold
 - Italic
 - Strikethrough
 - Underline
 5. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Set Diagram Trace Step Filtering Preferences

Step filtering when enabled, prevents the suspension of the application execution in a named Action Block or Event Action.

When Step filtering is enabled for an event type, all events of that type are filtered, regardless of the action block on which the filter is being executed. When step filtering is enabled for a named Action Block or Event Action, the Diagram Trace Utility filters the specified Action Block or Event Action by name.

When Step filtering is disabled, it has no effect on action blocks or event actions.

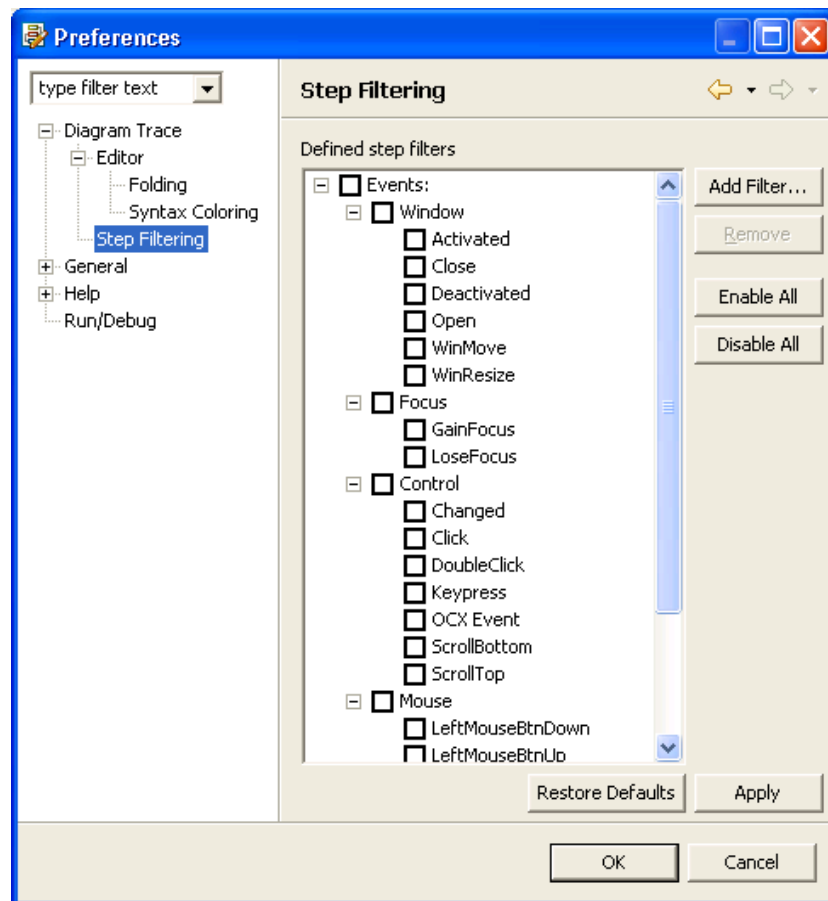
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to Diagram Trace, Step Filtering.

The Step Filtering Preferences page appears.



Note: Click Enable All or Disable All to enable step filtering for all events or disable step filtering for all events.

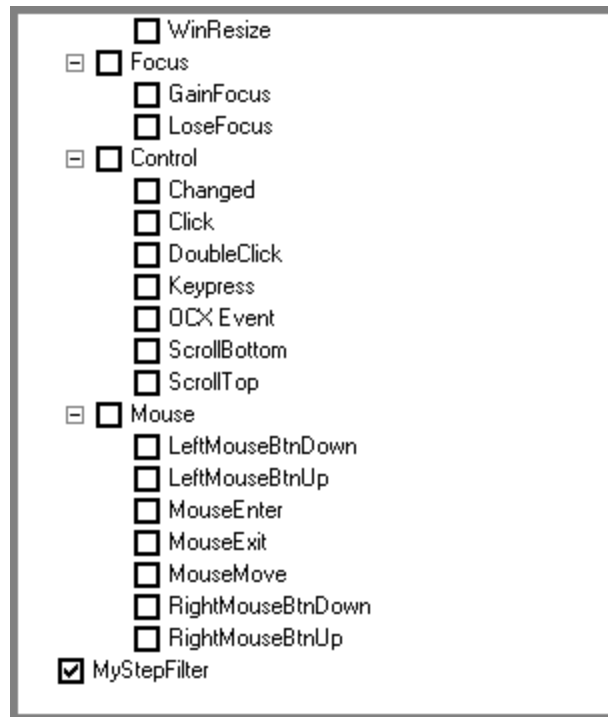
3. (Optional) Click Add Filter to add a step filter.

The Add Step Filter dialog opens.

Enter a value in the Pattern to filter field and click OK.

The new filter appears at the bottom of the list and is enabled by default.

Example: The following illustration shows a new enabled filter named MyStepFilter that is added at the bottom of the list.



4. (Optional) Select an event and click Remove.

The event is removed from the list.

5. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Set General Preferences

You can set general preferences to control the cycling of editor and view dialogs when displaying next and previous views or editors and to control the double-click and single-click mouse behaviors.

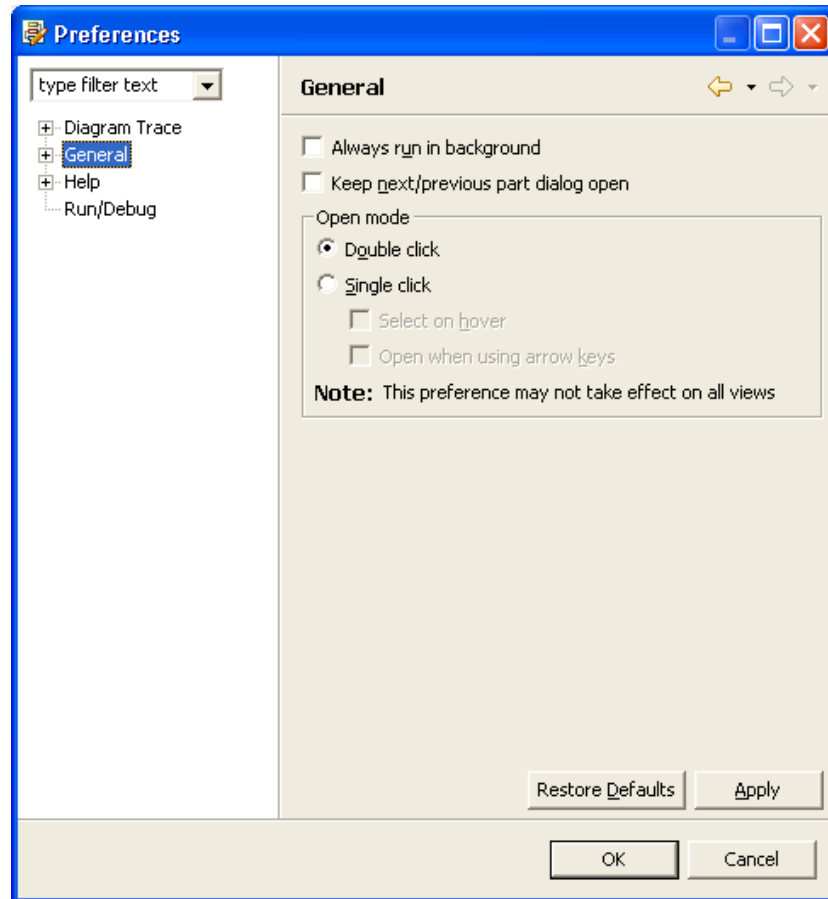
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Click General.

The General preferences page appears.



3. Change the default settings for the following fields to suit your needs:

Keep next/previous part dialog open

Specifies that the PAD Code View editor and the view cycle dialogs remain open after you release their activation keys. Usually, the dialogs or menus close as soon as you release the activation keys (such as CTRL+F7).

Default: Off

Example: When this preference is checked, you can use the key combination Ctrl +F7 on the Diagram Trace Utility main window to keep the pop-up menu listing views that can be opened to remain visible when you release the control keys. When this check box is cleared, the pop-up menu remains visible only as long as you hold down the control key. If you release the control key, the pop-up menu closes.

Open Mode

Specifies the default behavior for opening views and PAD Code View editors. You can select one of the following methods for opening these resources:

- Double click—Specifies a single-click to select a resource and a double-click to open that resource in an editor.
- Single click (Select on hover)—Specifies hovering the mouse over a resource to select it and single-click on the resource to open it in an editor.
- Single click (Open when using arrow keys)—Specifies using the arrow keys to select a resource and automatically open it in an editor.

Default: Double click

Note: The effect of these selections may vary depending on the view that has the focus.

4. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Note: The Diagram Trace Utility does not support the field Always run in background.

Set General Appearance Preferences for Editor and View Tabs

The General Appearance preferences settings specify the way PAD Code View editor tabs and view tabs are positioned, the style of tabs that are used, and whether animation is used to animate views to their location when they are opened or closed.

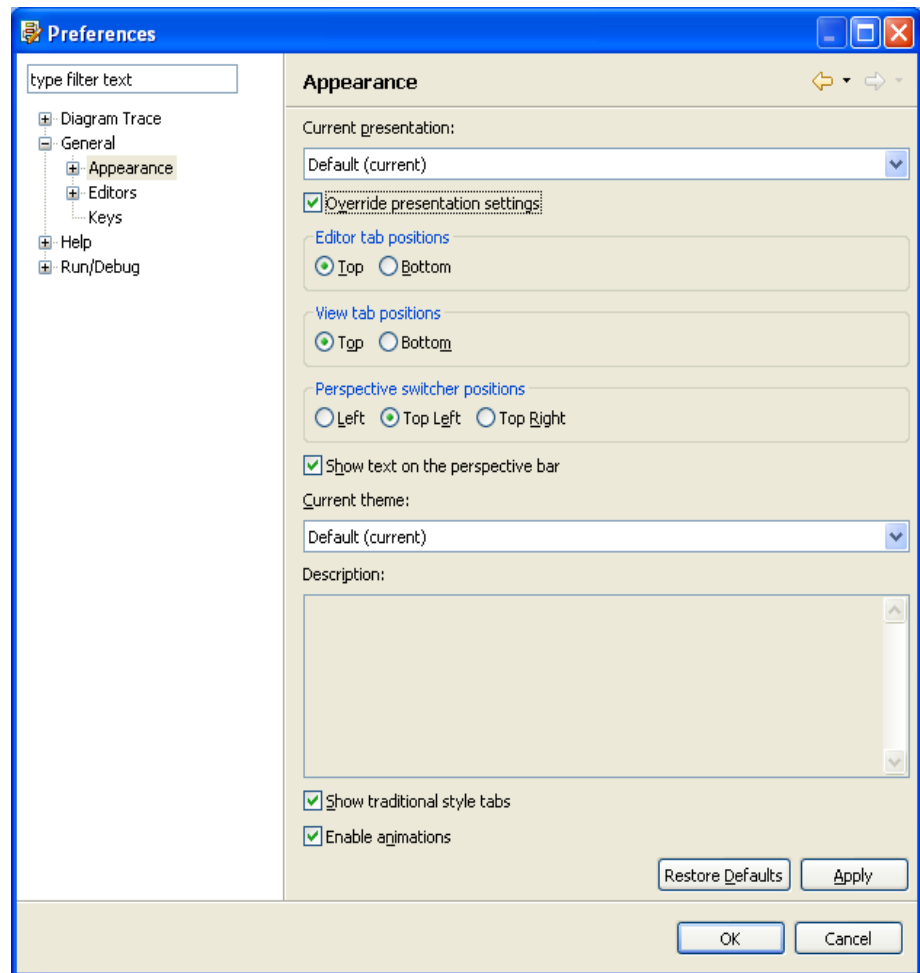
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Appearance.

The Appearance preferences page appears.



3. Change the default settings for the following fields to suit your needs:

Override Presentation Settings

Locally override the settings from the current presentation's defaults.

Default: Disabled

Editor tab position

Changes the location of the tabs in the PAD Code View editor.

Default: Top

View tab positions

Change the location of tabs in all views.

Default: Top

Perspective switcher positions

Not supported

Show traditional style tabs

Changes the display of traditional style tabs (square tabs) to curved tabs.

Default: On

Enable animations

Enable/disable the feature where views animate to their location when closed or opened.

Default: Enabled

Note: The Diagram Trace Utility does not support the Perspective switcher positions, Current Presentation, Current theme, or Show text on the perspective bar fields, or Enable animations options.

4. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Set General Appearance Preferences for Colors and Fonts

The General Appearance preference settings for colors and fonts specify the colors and fonts that are used for basic items such as hyperlinks, text, and error message text, debug fonts, and the colors and fonts to use for various components of views and editors.

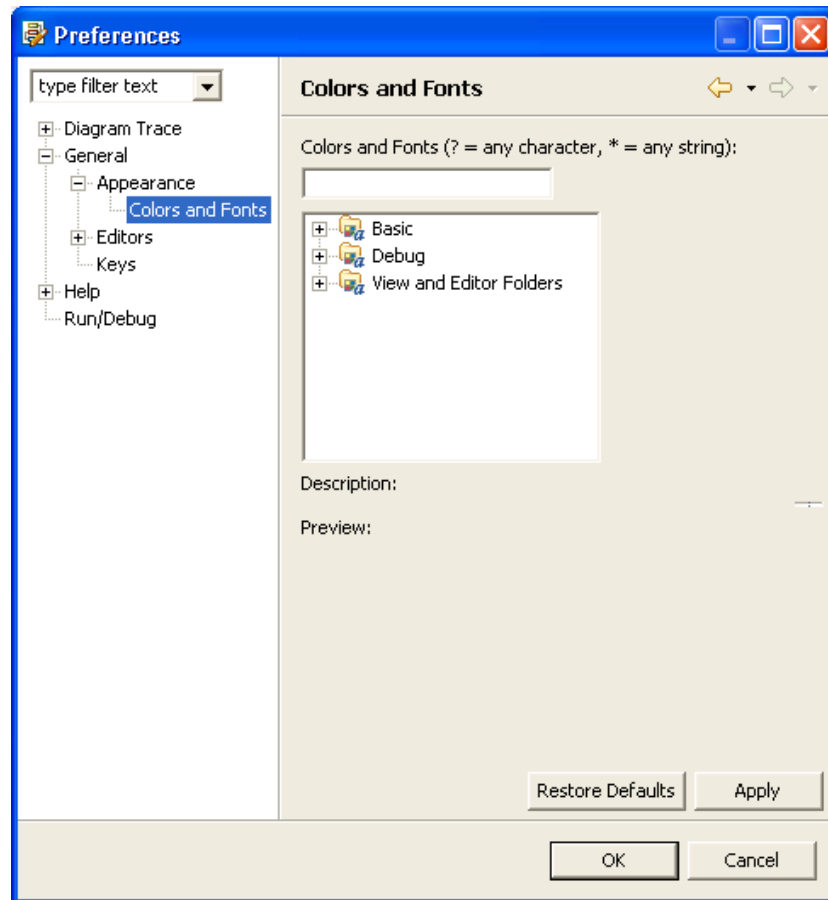
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Appearance, Colors and Fonts.

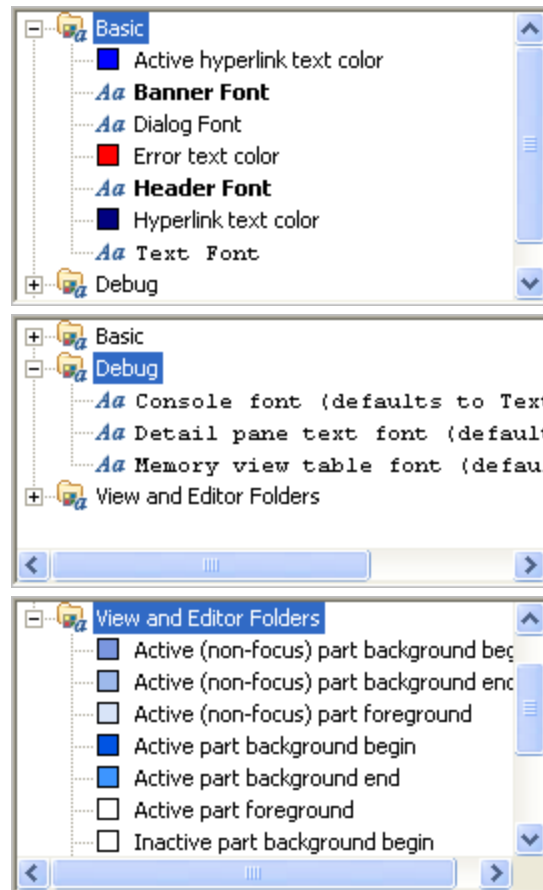
The Colors and Fonts Preferences page appears.



3. Click the plus sign (+) beside the Basic, Debug, and View and Editor Folders to open the folders.

The default color and font settings for different components in each folder appear.

The following illustrations show the default settings in the Basic, Debug, and View and Editor folders:



4. Select the item you want to change and click the Color button or click Change.

The Color or Font selection dialog appears.

Note: For font selections, you can click Use System Font to let the operating system automatically set the font to a reasonable value. For example, on Windows systems, the operating system sets the font selected in the Display Properties control panel.

5. Select a Color or font from the Color or Font selection dialog and click OK.

The Color or Font selection dialog closes.

6. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Set General Preferences for PAD Code View Editors

The General Preference settings for editors affect the general appearance and behavior of the PAD Code View editors.

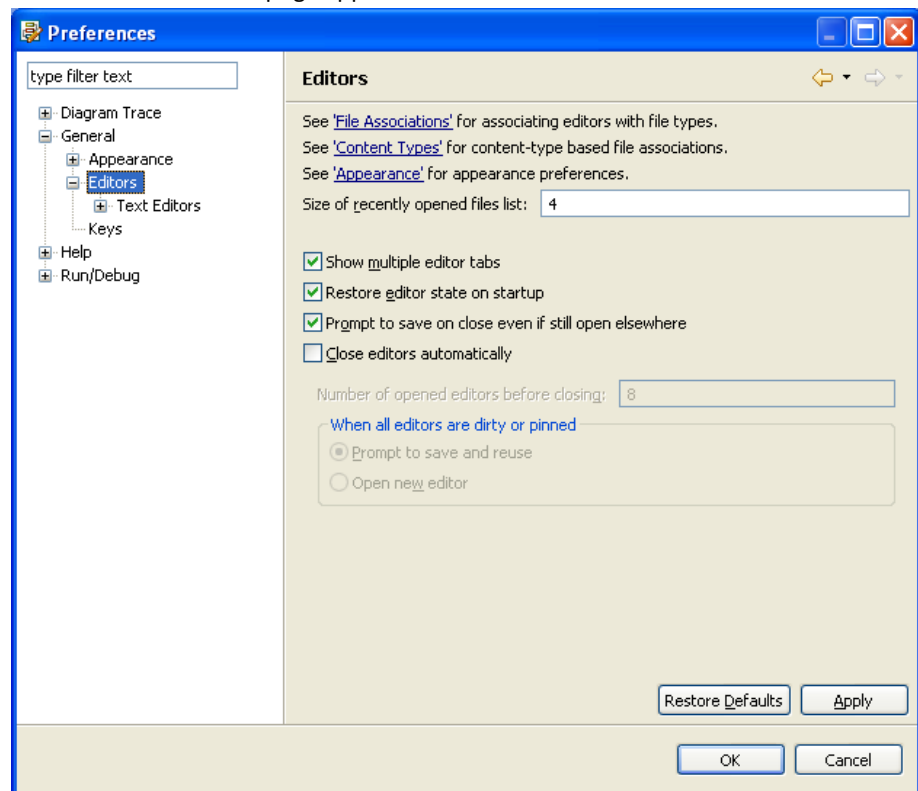
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Editors.

The Editors Preference page appears.



3. Enter values for the following fields to suit your needs:

Size of recently opened files

Controls the number of files displayed in the File menu.

Default: 4

Limits: 0 to 15

Show multiple editor tabs

Display of multiple PAD Code View editor tabs.

Default: On

Note: When this option is Off, editor workbooks have one large tab and all PAD Code View editors that are not visible are available only from the drop-down menu.

Restore editor state on startup

Indicates if open editors will be restored when starting up DTU.

Default: On

Prompt to save on close even if still open elsewhere

Default: On

Close editors automatically

Disables reusing editors in the Diagram Trace Utility.

Default: Off

Number of editors to use before recycling

Displays the number of PAD Code View editors you can use before recycling.

Default: 8

Note: This option is enabled when you select Close editors automatically.

When all editors are dirty or pinned

Prompts a dialog to save and reuse editors or opens a new PAD Code View editor when all editors are dirty or pinned. If turned on, the Pin editor action is added to the PAD Code View editor toolbar and editor tab menu.

Default: Prompt to save and reuse

Note: This option is enabled when you select Close editors automatically.

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Note: The Diagram Trace Utility does not support use of the 'Prompt to save on close even if still open elsewhere' field.

Set General Preferences for Text Editors

The text editor preferences settings affect the display and behavior of PAD Code View text editors.

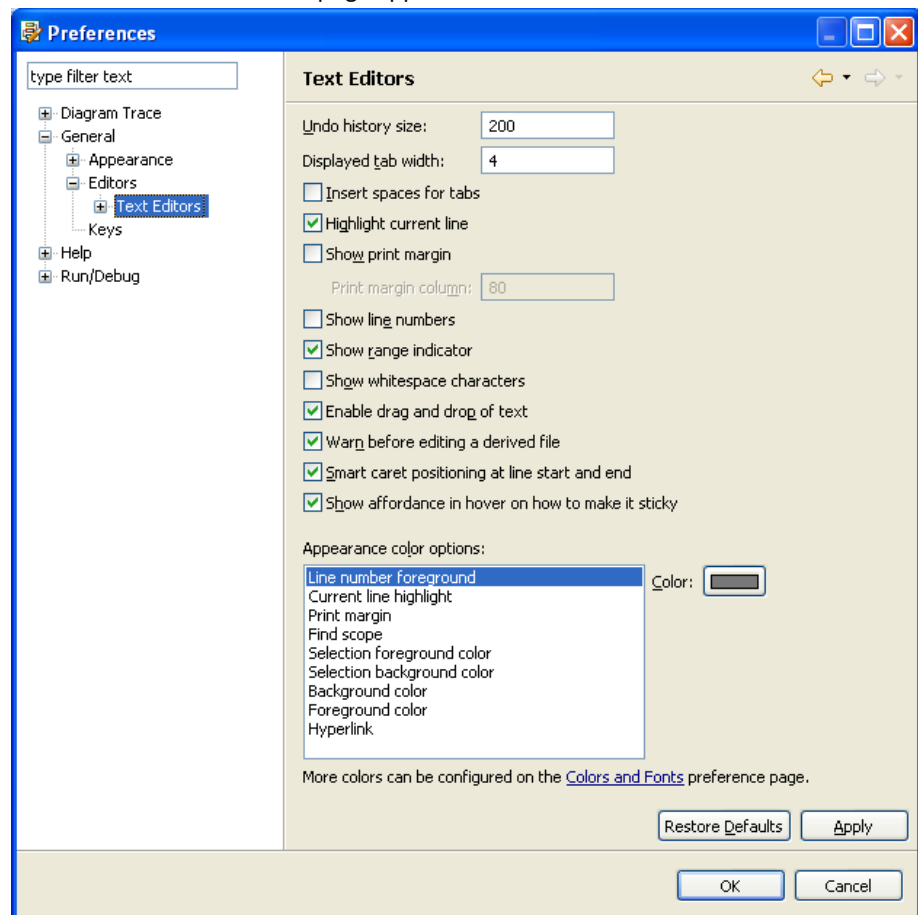
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Editors, Text Editors.

The Text Editors Preference page appears.



3. Enter values for the following fields to suit your needs:

Highlight current line

Highlights the current line in text editors.

Default: On

Show print margin

Displays the print margin.

Default: Off

Print margin columns

Lets you specify the print margin column.

Default: 80

Note: This field is enabled if you select Show print margin.

Show line numbers

Displays line numbers on the left side of text editors.

Default: Off

Appearance color options

Lets you specify the color for the following values:

- Line number foreground
- Current line highlight
- Print margin
- Find Scope
- Selection foreground color
- Selection background color
- Background color
- Foreground color
- Hyperlink

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Note: The Diagram Trace Utility does not support use of the Undo history size, Displayed tab width, Insert spaces for tabs, Show range indicator, Enable drag and drop of text, Warn before editing a derived file, Smart caret positioning at line start and end, or Show affordance in hover on how to make it sticky. Also, the Diagram Trace Utility does not support the Selection foreground color option or Find scope in the Appearance color options list.

Set Accessibility Preferences for Text Editors

The Accessibility Preferences settings affect the use of different carets in text editors.

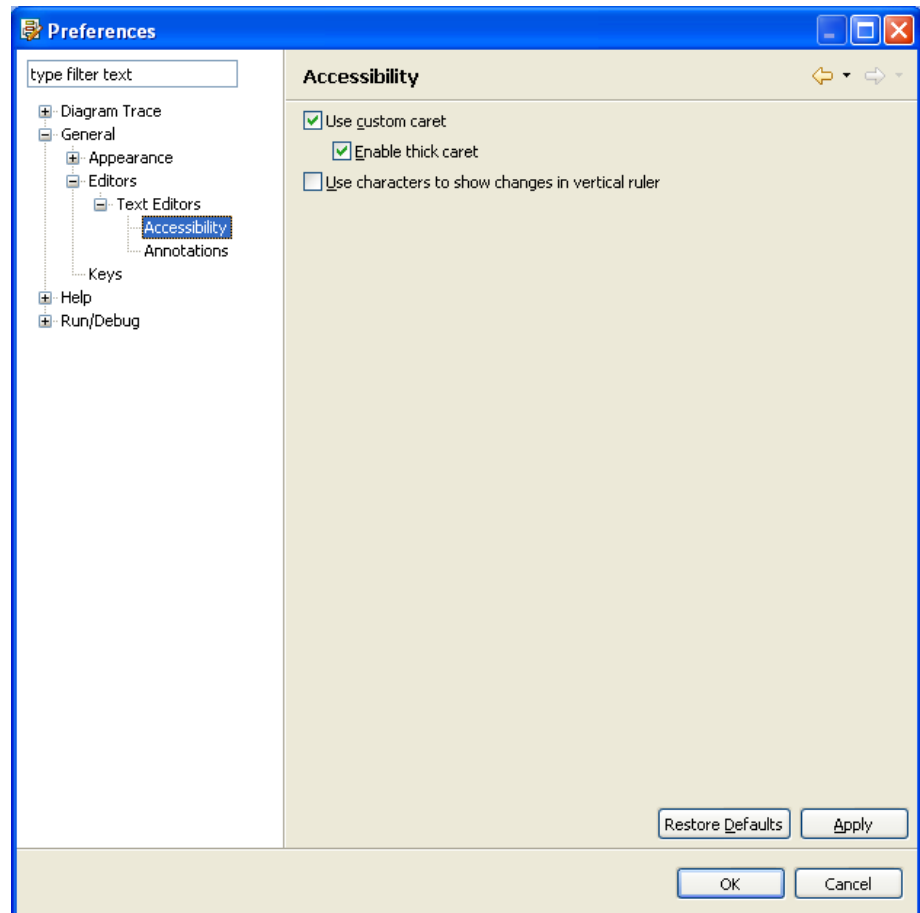
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Editors, Text Editors, Accessibility.

The Accessibility Preferences dialog appears.



3. Set the following options to suit your needs:

Use custom caret

Uses the default custom caret for Overwrite and Insert modes.

Default: On

Enable thick caret

Specifies thick caret for Overwrite and Insert modes.

Default: On

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Note: The Diagram Trace Utility does not support the Use characters to show changes in vertical ruler option.

Set Annotations Preferences for Text Editors

Annotation preference settings affect the appearance and location in which different annotations such as bookmarks and breakpoints are displayed in PAD Code View editors.

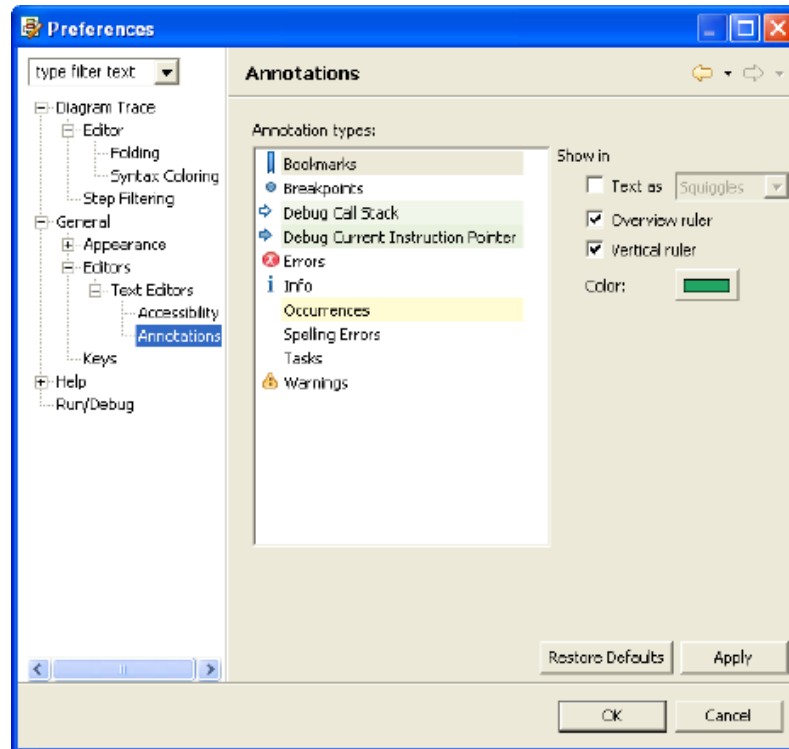
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Editors, Text Editors, Annotations.

The Annotations Preferences dialog appears.



3. Set the following options to suit your needs:

Annotations

Displays the default setting for the annotation selected.

Note: The Diagram Trace Utility does not support the Annotation types Spelling Errors or Tasks.

Show in

Specifies the location in which the annotation appears (in text with selected attributes, in the Overview ruler, or in the Vertical ruler) in the PAD Code View editor.

Color

Lets you specify a basic or custom color for the selected annotation from the Color selection dialog.

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Set General Preferences for Keys

The General Preference settings for keys affect the key mappings or bindings used to perform various functions in the Diagram Trace Utility.

Note: The Diagram Trace Utility contains a key mappings feature, which provides the ability to modify key assignments. This feature is inherited from the Eclipse Platform. The functionality of these key mappings has not been certified.

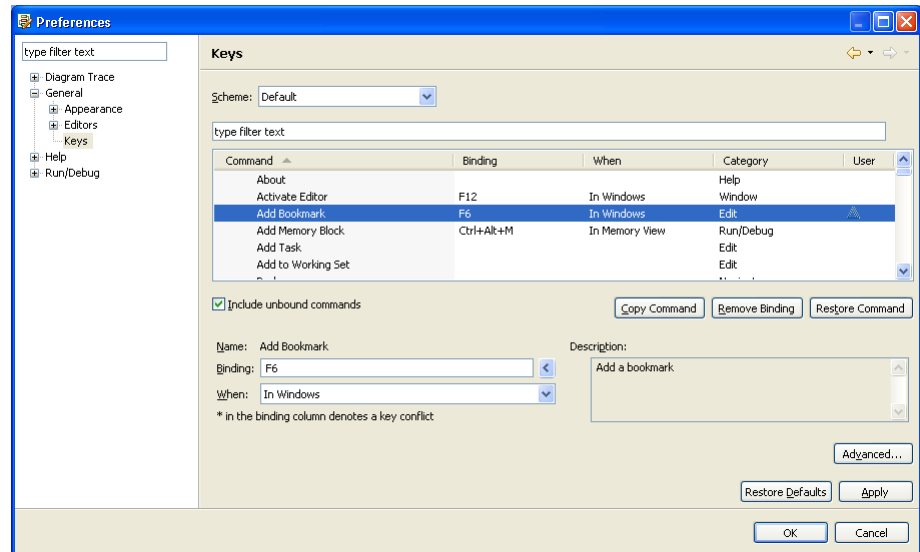
Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Keys.

The Keys Preferences appears in the right pane and displays a list of the default key mappings for all commands and functions.



3. Select a command key sequence you want to change.
4. If there is an existing command sequence, select Remove Binding.
5. Type the key sequence you want to assign to the function in the Binding field.

Note: If the command key sequence already exists, an asterisk (*) is placed next to the key in the Binding column.

6. Select the options from When drop-down.

7. Click OK.

The Diagram Trace Utility saves the new setting and closes the Preferences dialog.

Note: You can restore the default settings for the command by selecting the command and clicking Restore Command. To restore all keys to the default settings, Click Restore Defaults.

Export Key Preferences to a File

You can export the key mapping preferences to an Excel spreadsheet and save it in an Excel comma separated value (*.csv) file format.

Note: The Diagram Trace Utility contains a key mappings feature, which provides the ability to modify key assignments. This feature is inherited from the Eclipse Platform. The functionality of these key mappings has not been certified.

Follow these steps:

1. Select Preferences from the Window menu.

The Preferences dialog appears.

2. Navigate to General, Keys and select the View tab.

The Keys Preferences View tab appears in the right pane and displays a list of the default key mappings for all commands and actions.

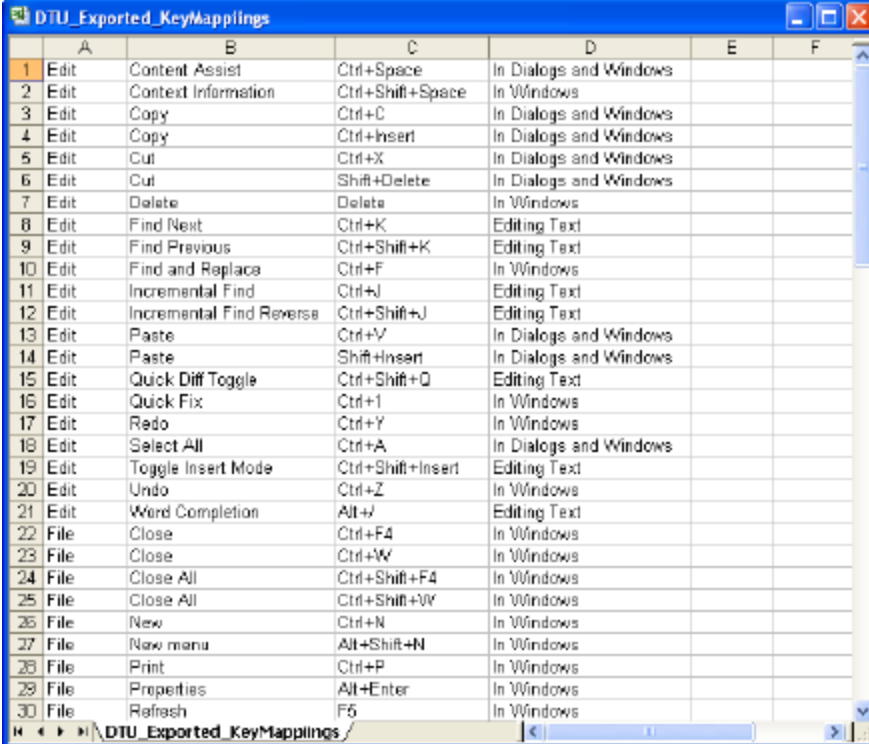
3. Click Export.

The Windows Save As dialog appears.

4. Specify a file name and location for the exported file and click Save.

The exported key mappings are saved to a file in an Excel Comma Separated Values (*.csv) formatted file in the location you specified.

Example: The following illustration shows an exported key mappings file:



	A	B	C	D	E	F
1	Edit	Content Assist	Ctrl+Space	In Dialogs and Windows		
2	Edit	Context Information	Ctrl+Shift+Space	In Windows		
3	Edit	Copy	Ctrl+C	In Dialogs and Windows		
4	Edit	Copy	Ctrl+Insert	In Dialogs and Windows		
5	Edit	Cut	Ctrl+X	In Dialogs and Windows		
6	Edit	Cut	Shift+Delete	In Dialogs and Windows		
7	Edit	Delete	Delete	In Windows		
8	Edit	Find Next	Ctrl+K	Editing Text		
9	Edit	Find Previous	Ctrl+Shift+K	Editing Text		
10	Edit	Find and Replace	Ctrl+F	In Windows		
11	Edit	Incremental Find	Ctrl+J	Editing Text		
12	Edit	Incremental Find Reverse	Ctrl+Shift+J	Editing Text		
13	Edit	Paste	Ctrl+V	In Dialogs and Windows		
14	Edit	Paste	Shift+Insert	In Dialogs and Windows		
15	Edit	Quick Diff Toggle	Ctrl+Shift+Q	Editing Text		
16	Edit	Quick Fix	Ctrl+I	In Windows		
17	Edit	Redo	Ctrl+Y	In Windows		
18	Edit	Select All	Ctrl+A	In Dialogs and Windows		
19	Edit	Toggle Insert Mode	Ctrl+Shift+Insert	Editing Text		
20	Edit	Undo	Ctrl+Z	In Windows		
21	Edit	Word Completion	Alt+/	Editing Text		
22	File	Close	Ctrl+F4	In Windows		
23	File	Close	Ctrl+W	In Windows		
24	File	Close All	Ctrl+Shift+F4	In Windows		
25	File	Close All	Ctrl+Shift+W	In Windows		
26	File	New	Ctrl+N	In Windows		
27	File	New menu	Alt+Shift+N	In Windows		
28	File	Print	Ctrl+P	In Windows		
29	File	Properties	Alt+Enter	In Windows		
30	File	Refresh	F5	In Windows		

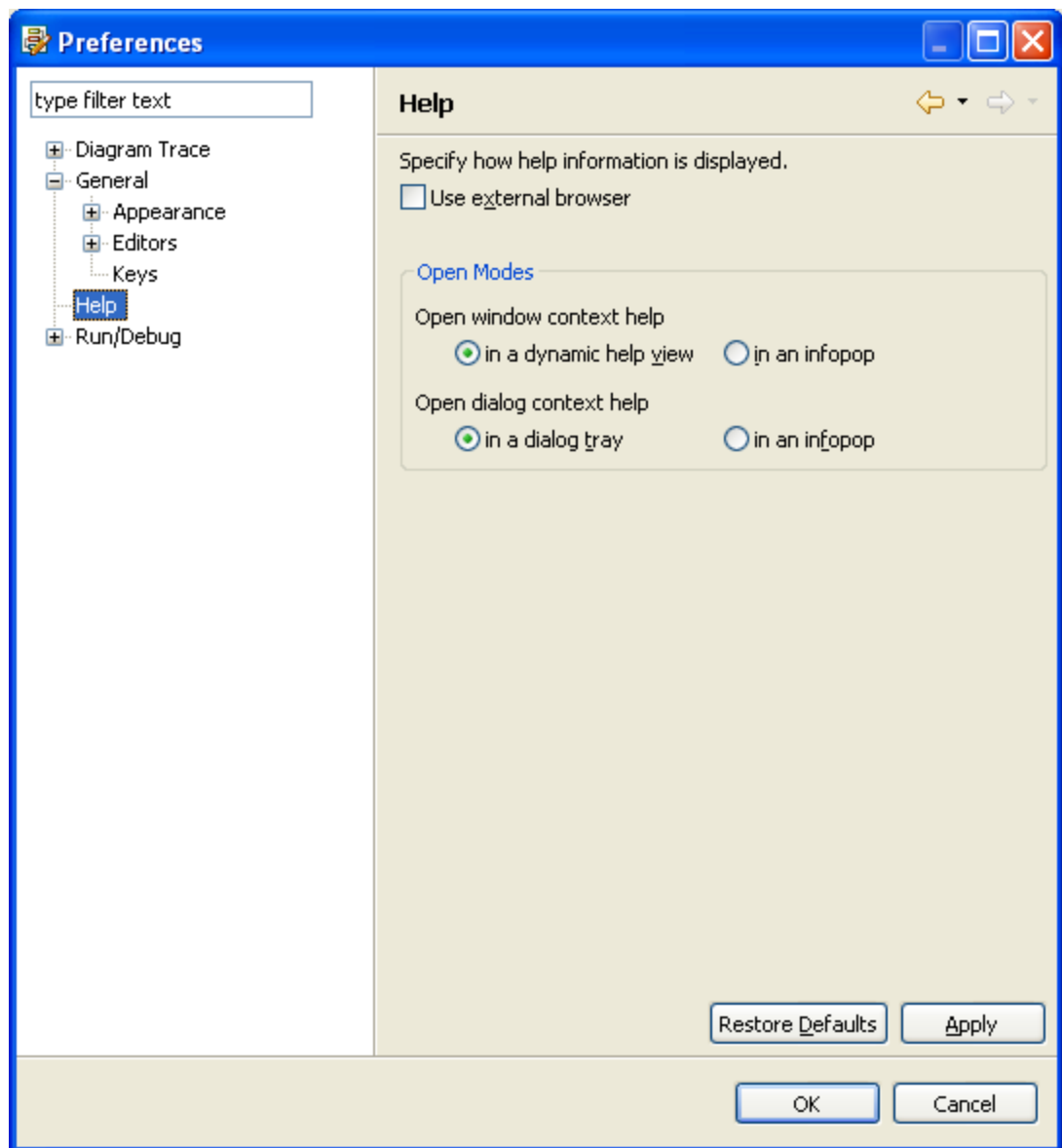
5. Click OK.

The Preferences dialog closes.

More information:

[Set General Preferences for Keys](#) (see page 131)

Set Help Preferences



The Help Preferences settings affect the way online help is displayed and determine whether an external browser is used to display help information.

Note: If you do not specify use of an external help browser, the Diagram Trace Utility uses an internal panel to host the web browser.

Follow these steps:

1. Select Preferences from the Window menu.
The Preferences dialog appears.

2. Navigate to Help

The Help Preferences dialog appears.

3. Set the following options to suit your needs:

Use external browser

Specifies the use of an external browser to display online help.

Default: Off

Open window context help

Specifies whether to open the context help for windows in a dynamic view panel or in a pop-up information box.

Default: in a dynamic help view

Open dialog context help

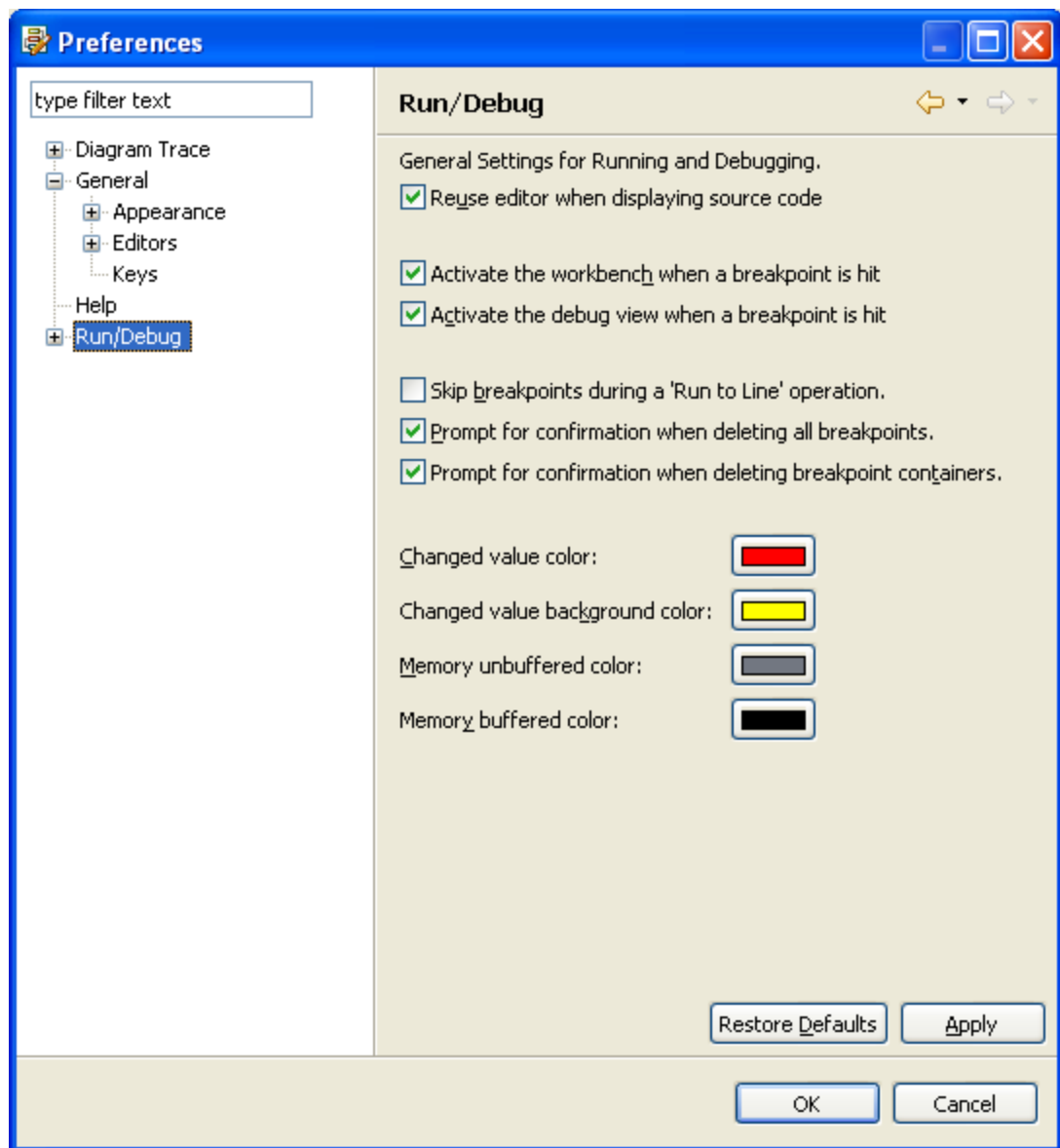
Specifies whether to open dialog context help in a dynamic help window or in a pop-up information box.

Default: in a dynamic help window

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Set Run/Debug Preferences



The Run/Debug preference settings affect the general behaviors for running a debug session.

Follow these steps:

1. Select Preferences from the Window menu.
The Preferences dialog appears.
2. Navigate to Run/Debug.
The Run/Debug Preferences dialog appears.

3. Set the following options to suit your needs:

Reuse editor when displaying PAD code

Specifies the debugger to reuse the PAD Code View editor it opened to display the PAD code from different PAD files. This prevents the debugger from opening an excessive number of PAD Code View editors.

Default: On

Skip breakpoints during a 'Run to Line' operation

Allows the debugger to ignore any breakpoints and continue execution when it is executing a Run to Line operation.

Variables View changed value color

Lets you reset the color that the debugger displays for changed values in the Variables view.

Default: Red

Note: The Diagram Trace Utility does not use the remaining options on this dialog.

4. Click OK.

The Diagram Trace Utility saves the new settings and closes the dialog.

Chapter 7: Troubleshooting

This chapter describes how to troubleshoot errors using the logs available in Diagram Trace Utility.

This section contains the following topics:

[Console and Error Logs](#) (see page 139)

[Review Contents of the Console Log View](#) (see page 139)

[Review Contents of the Error Log View](#) (see page 141)

[Troubleshoot Connection Problems](#) (see page 145)

[Troubleshoot Configuration Problems](#) (see page 146)

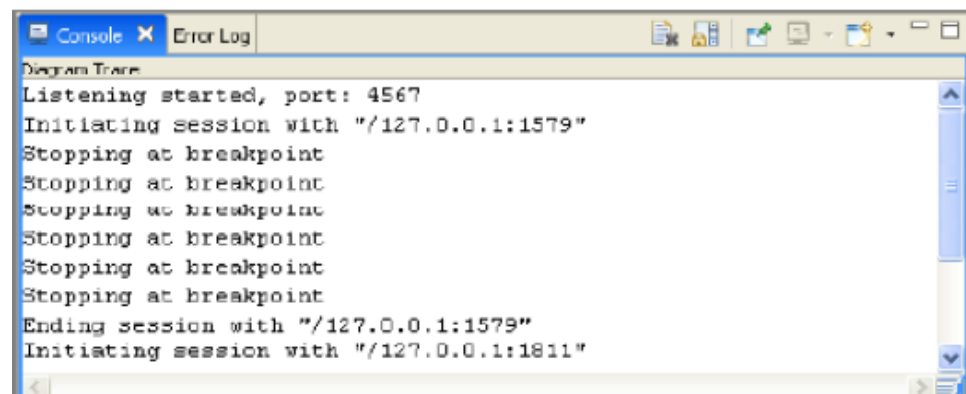
Console and Error Logs

Logs are one of the first places to look when you try to diagnose any type of error. The Diagram Trace Utility provides the following views to help you identify errors and troubleshoot problems:

- [Console Log View](#) (see page 139)
- [Error Log View](#) (see page 141)

Review Contents of the Console Log View

The Diagram Trace Utility logs all standard output in the Console view, which is a read-only view. You can review the contents of all the messages in the Console view during a debugging session. The following illustration shows sample messages logged in the Console view:



You can also search for specific text in the Console view.

To search for text in the Console view

Follow these steps:

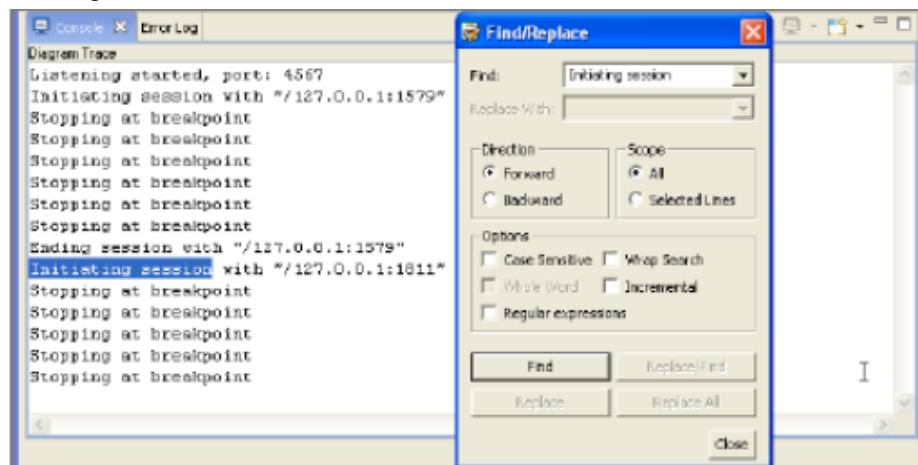
1. Right-click inside the viewing area of the Console view and select Find/Replace from the pop-up menu.

The Find/Replace dialog appears.

2. Enter the word or phrase you want to find and select any other options, such as the direction or scope of the search, and click Find.

The Diagram Trace Utility finds and highlights the text you specified in the Console View.

Example: The following illustration shows a search initiated to find the phrase *Initiating session*:



Note: The Console view is read-only and does not support the Replace/Find, Replace, or ReplaceAll options.

3. (Optional) Click Find again to find the next occurrence of the specified text.

The Diagram Trace Utility finds and highlights the next occurrence of the specified text in the Console View.

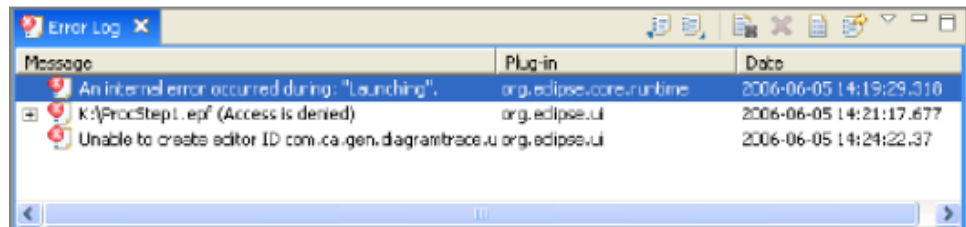
Note: You can type a different word or phrase in the Find box and repeat Steps 1 and 2 to find the occurrences of the new text.

4. Click Close.

The Find/Replace dialog closes.

Review Contents of the Error Log View

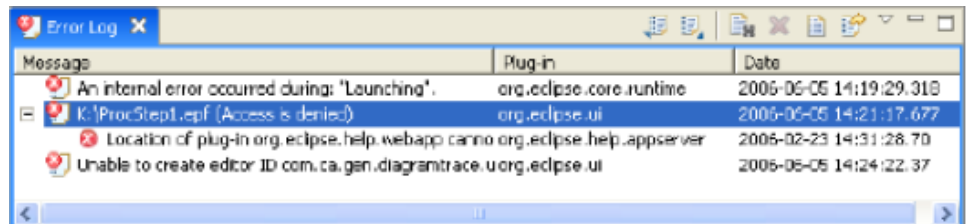
The Error Log view captures internal errors that are caused by the operating system or the DiagramTrace Utility. The following illustration shows a sample Error Log view:



The Error Log view displays the following information for each error:

- The status of the error (for example, error or warning)
- A brief description of the error
- The plug-in from which the error originated
- The date and time that the error occurred

Note: If a plus sign (+) precedes an error, the error is a complex problem. This indicates that a number of errors contribute to the problem. You can double-click the error to expand and view the list of errors as shown in the following example:



View Error Details

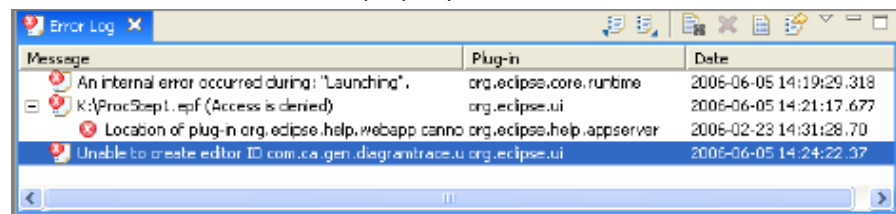
You can view the error details for the errors logged in the Error Log view.

Follow these steps:

1. In the Error Log view, click the plus sign to the left of the error you want to review.

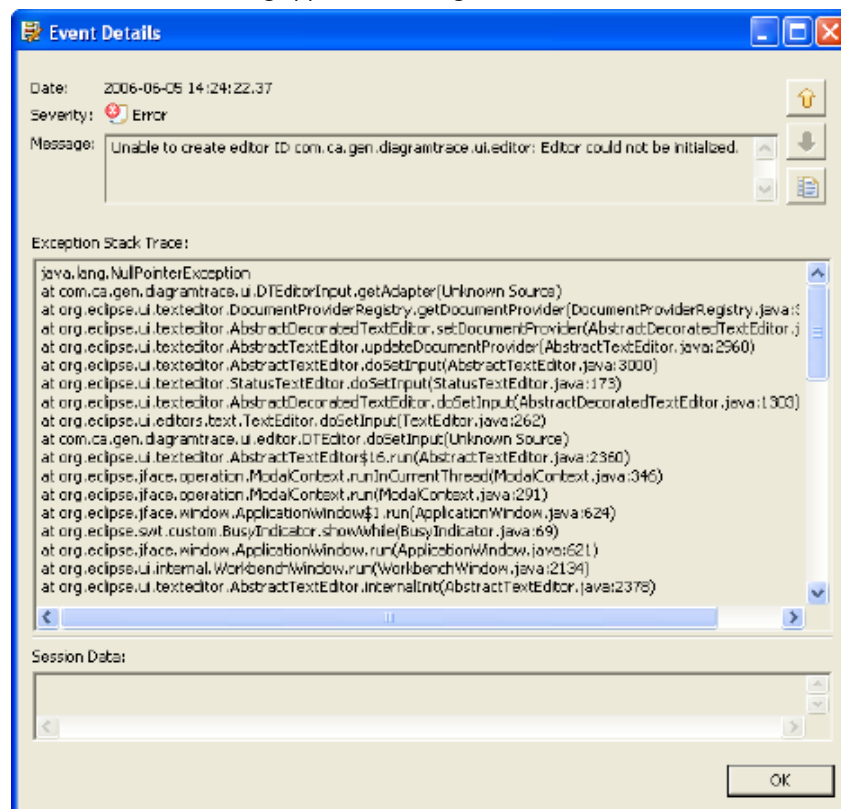
The list expands and displays individual errors.

Example: The following illustration shows the individual errors for the error —Help document could not be indexed properly.



2. Double-click the selected entry in the Error log view

The Event Details dialog appears showing more details about the error.



- Click OK after you finish reviewing the event details.

The Event Details dialog closes.

View the Detail Log for a Selected Error in Notepad

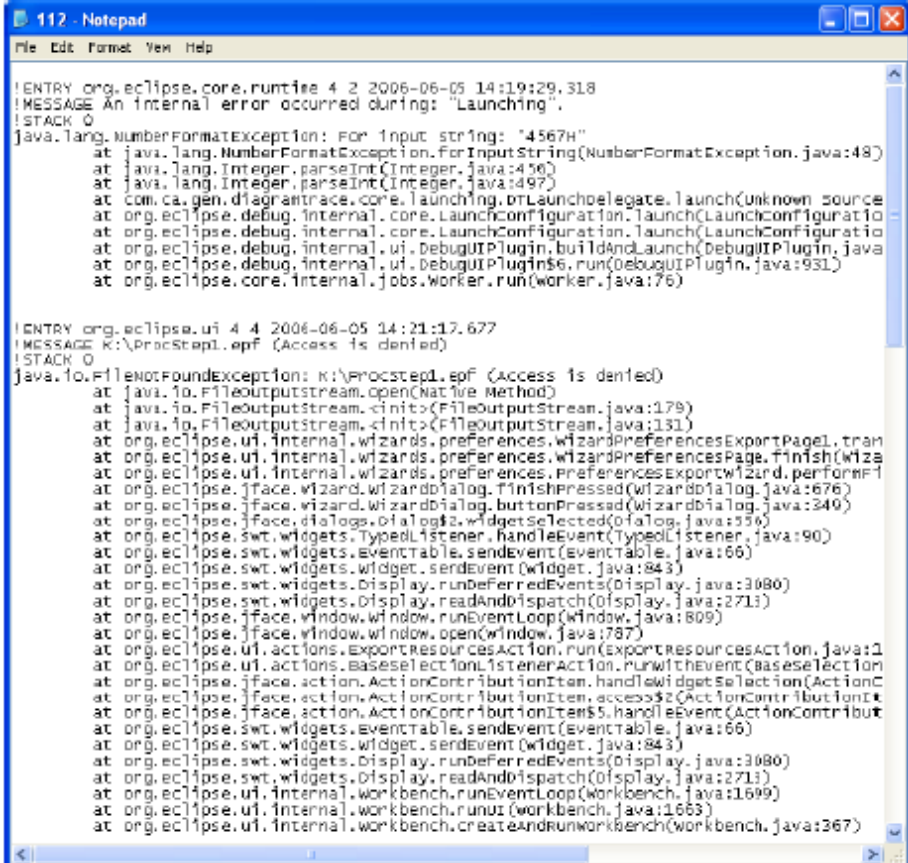
You can display details for any selected error in the Error Log view in Notepad.

Follow these steps:

- Select an error in the Error Log view to review and click the Open Log icon on the Error Log view toolbar (or select the Open Log option from the right-click pop-up menu).

A Notepad opens and displays the details for the selected error.

Example: The following illustration shows the error description for a sample error:



```

112 - Notepad
File Edit Format View Help

ENTRY org.eclipse.core.runtime 4 2 2006-06-05 14:19:29.318
MESSAGE An internal error occurred during: "Launching".
STACK 0
java.lang.NumberFormatException: For input string: "4567H"
    at java.lang.NumberFormatException.forInputString(NumberFormatException.java:48)
    at java.lang.Integer.parseInt(Integer.java:456)
    at java.lang.Integer.parseInt(Integer.java:497)
    at com.ca.gen.diagrantrace.core.launching.OTLaunchDelegate.launch(Unknown Source)
    at org.eclipse.debug.internal.core.LaunchConfiguration.launch(LaunchConfiguration.java:100)
    at org.eclipse.debug.internal.ui.DebugUIPlugin.buildAndLaunch(DebugUIPlugin.java:100)
    at org.eclipse.debug.internal.ui.DebugUIPlugin$6.run(DebugUIPlugin.java:931)
    at org.eclipse.core.internal.jobs.Worker.run(Worker.java:76)

ENTRY org.eclipse.ui 4 4 2006-06-05 14:21:17.677
MESSAGE K:\ProcStep1.epf (Access is denied)
STACK 0
java.io.FileNotFoundException: K:\ProcStep1.epf (Access is denied)
    at java.io.FileOutputStream.open(Native Method)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:179)
    at java.io.FileOutputStream.<init>(FileOutputStream.java:131)
    at org.eclipse.ui.internal.wizards.preferences.WizardPreferencesExportPage1.tram
    at org.eclipse.ui.internal.wizards.preferences.WizardPreferencesPage.finish(Wiza
    at org.eclipse.ui.internal.wizards.preferences.PreferencesExportWizard.performF
    at org.eclipse.jface.wizard.WizardDialog.finishPressed(WizardDialog.java:676)
    at org.eclipse.jface.wizard.WizardDialog.buttonPressed(WizardDialog.java:349)
    at org.eclipse.jface.dialogs.Dialog$2.widgetSelected(Dialog.java:556)
    at org.eclipse.swt.widgets.TypedListener.handleEvent(TypedListener.java:90)
    at org.eclipse.swt.widgets.EventTable.sendEvent(EventTable.java:66)
    at org.eclipse.swt.widgets.Widget.sendEvent(Widget.java:843)
    at org.eclipse.swt.widgets.Display.runDeferredEvents(Display.java:3080)
    at org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:2713)
    at org.eclipse.jface.window.Window.runEventLoop(Window.java:809)
    at org.eclipse.jface.window.Window.open(Window.java:787)
    at org.eclipse.ui.actions.ExportResourcesAction.run(ExportResourcesAction.java:1
    at org.eclipse.ui.actions.BaseSelectionListenerAction.runWithEvent(BaseSelection
    at org.eclipse.jface.action.ActionContributionItem.handleWidgetSelection(ActionC
    at org.eclipse.jface.action.ActionContributionItem.access$2(ActionContributionI
    at org.eclipse.jface.action.ActionContributionItem$5.handleEvent(ActionContrib
    at org.eclipse.swt.widgets.EventTable.sendEvent(EventTable.java:66)
    at org.eclipse.swt.widgets.Widget.sendEvent(Widget.java:843)
    at org.eclipse.swt.widgets.Display.runDeferredEvents(Display.java:3080)
    at org.eclipse.swt.widgets.Display.readAndDispatch(Display.java:2713)
    at org.eclipse.ui.internal.workbench.runEventLoop(Workbench.java:1699)
    at org.eclipse.ui.internal.workbench.runUI(Workbench.java:1663)
    at org.eclipse.ui.internal.workbench.createAndRunWorkbench(Workbench.java:367)
  
```

- Select Exit from the File menu after you finish reviewing the details log.

The Notepad closes.

Import an Error Log

You can import an existing error log to review from your local drive or network drive.

Follow these steps:

1. Click the Import Log icon on the Error Log view toolbar.

The Windows Open dialog appears.

2. Enter the file name of the error log in the File Name field or use the browser to locate the file.

The imported file appears in the Error Log view.

Note: The imported file must adhere to the Error Log file format to be displayed correctly in the Error Log view.

Export an Error Log

You can export an error log to your local drive or network drive.

Follow these steps:

1. Click the Export Log icon on the Error Log View toolbar.

The Windows Save As dialog appears.

2. Enter a file name in the File Name field and specify the location for saving the file on your local system or network drive. Click Save.

The exported error log file is saved in the specified location.

Clear and Restore an Error Log

You can clear the contents of the Error Log view or restore the contents to the Error Log view.

To clear the contents of the Error Log view, click the Clear Log Viewer icon on the Error Log View toolbar. This action removes the data from the Error Log view but does not remove the Error Log file.

To restore the content of the Error Log view, click the Restore Error Log icon on the Error Log view toolbar.

Troubleshoot Connection Problems

If you experience problems connecting to the Diagram Trace Utility server, perform the following tasks:

- Verify that TCP/IP is properly installed on your workstation. Try to ping your Diagram Trace Utility server machine to verify that it is visible in the network. If you are unsure of how to configure TCP/IP, call your local systems administrator.
- If you specified the server address by name, try entering the decimal IPv4 address instead (for example, 1.1.1.1) or hexadecimal IPv6 address (for example FE80:0:0:1::1). Your machine may not be configured properly to work with your local name server.
- Verify the application being traced has been properly enabled and configured for tracing. The application must have been generated with the Trace option enabled and have the host name set to the machine on which the Diagram Trace Utility is installed. Also, the port number configured in the application must match the port number configured in the Diagram Trace Utility.
- Ping the machine that is running the Diagram Trace Utility from the machine that will be communicating with the Diagram Trace Utility (either the server machine or another workstation). If your ping is unsuccessful, the server (or workstation) does not know the address of the machine running the Diagram Trace Utility. You may need to establish Domain Name Services (DNS) on the server (or workstation) in order to recognize the machine running the Diagram Trace Utility.
- Verify that you do not have a firewall issue. If you are generating a trace log file on your server (using the `-t` flag), you may see the following text:

Failed to connect to server. Response code 78.

This is a timeout response code. One possible reason for this is a firewall restriction.

To verify if you have a firewall issue:

Follow these steps:

1. Start the Diagram Trace Utility with default port number or port number of choice.
2. From your server (or alternate workstation), enter the following request:

```
telnet ip_address port_number
```

ip_address

Defines the IP address of the machine running the Diagram Trace Utility.

port_number

Defines the port number used when starting the Diagram Trace Utility.

If you can reach this port number successfully, you will see something like the following (using 172.24.12.127 as the IP address and 4567 as the port number in this example):

```
telnet 172.24.12.127 4567
Trying...
Connected to 172.24.12.127.
```

3. Stop the Diagram Trace Utility.
4. From your server (or alternate workstation), enter the same telnet request again.

Since the port number has been disabled (no longer listening), you will now see something like the following:

```
telnet 172.24.12.127 4567
Trying...
telnet: connect: A remote host refused an attempted connect operation.
```

Note: You might receive something like the following:

```
telnet 172.24.12.127 4567
Trying...
```

Which means that you are not able to communicate with the machine running the Diagram Trace Utility.

This could indicate that you have a firewall issue. If you have a firewall running, make appropriate changes to allow traffic through the port number being used to communicate with the Diagram Trace Utility.

Troubleshoot Configuration Problems

If you encounter configuration problems or receive start-up error messages, type the following command from a command prompt window:

```
"%GENxx%\gen\dt\DiagramTrace.exe" -clean
```

xx

Identifies the current release number of CA Gen.

This command deletes runtime metadata and cached data in the configuration directory.

Appendix A: Using Accessibility Keys

You can navigate the Diagram Trace Utility main window using certain keyboard keys or key combinations. For example, you can use the Tab key to move to controls (fields, buttons, check boxes, and so on) in a particular dialog or view and its related icons.

To navigate to the main controls in the Diagram Trace Utility main window or to tab out of views that use the Tab key (such as PAD Code View editors) use the Ctrl+Tab key combination.

This section contains the following topics:

[Access the Menus](#) (see page 147)

[Access the Controls](#) (see page 148)

[Switch Between PAD Code View Editors and Views](#) (see page 148)

[Set the Color Preferences](#) (see page 148)

[Navigate the Online Help System](#) (see page 149)

[Change the Help Browser Settings](#) (see page 150)

Access the Menus

Most menus have assigned mnemonics or accelerator keys for selecting menu options and submenu options. You can also use the arrow keys to move to and select options on menus and submenus.

You can access the various menus using the keys described in the following table:

Key or Key Combination	Description
F10	Accesses the menus on the main menu bar.
Shift+F10	Displays the pop-up menu for the current view. Note that this shortcut depends on your window manager, but typically it is Shift+F10.
Ctrl+F10	Opens the drop-down menu for the current view if one exists. For PAD Code View editors, Ctrl+F10 opens the menu for the vertical ruler on the left of the editor area.
Alt+mnemonic	Activates a menu on the Diagram Trace Utility main menu bar. For example, Alt+W activates the Window menu.
Alt	Gives the focus to the menu bar.

Access the Controls

Mnemonics are assigned to most control labels (for example, buttons, check boxes, and option buttons) in dialogs, preference pages, and property pages.

To access the control associated with a label, hold the Alt key down and press the underlined alphabet on the label.

Switch Between PAD Code View Editors and Views

The Diagram Trace Utility lets you activate switching using Ctrl key-function key combination. These key combinations let you switch between PAD Code View editors and views. All switch key combinations recall the last item you selected to provide rapid switching between two items.

Use the following switch key combinations to switch between a PAD Code View editor and a view:

- Ctrl+F6—Switches to editor
- Ctrl+F7—Switches to view

You can also use the following key combinations to activate PAD Code View editors and switch between open editors:

- Ctrl+E—Activates the PAD Code View editor or a drop-down menu containing a list of open editors
- Ctrl+PageUp and Ctrl+PageDown—Switches between the open editors

Set the Color Preferences

For color selection, the Diagram Trace Utility uses a dialog provided by the Windows operating system.

Follow these steps:

1. Select Preferences from the Window menu.
The Preferences dialog appears.
2. Navigate to General, Appearance, Colors and Fonts.
The Colors and Fonts Preferences page appears.

3. Select the required element whose color you want to change.

Example: You can select the color of Error Text by expanding the Basic tree view item.

A color button on the right side of the navigation tree becomes active. It shows the same color as the element selected.

4. Click the color button.

The Color dialog appears.

5. (Optional) Click Define Custom Colors and enter the basic colors you want to set using an HSL or RGB specification. Click Add to Custom Colors.

The color appears in the Custom Colors section.

6. Select any color from Basic colors or Custom colors. Click OK.

The Color dialog closes and the new color set is now visible on the color button.

7. Click OK.

The Diagram Trace Utility saves the changes and closes the Preferences dialog.

Navigate the Online Help System

You can navigate the online help system using the key combinations described in the following table.

Key or Key Combination	Description
Tab inside a frame (page)	Moves to the next link, button or topic node
Right Arrow/Left Arrow	Expands/collapses a tree node
Down Arrow or Tab	Moves to the next topic node
Up Arrow or Shift+Tab	Moves to the previous topic node
Enter	Selects a topic to display
Home or End	Scrolls all the way up or down
Alt+Left Arrow	Goes back
Alt+Right Arrow	Goes forward
Ctrl+Tab	Moves to next frame
Shift+Ctrl+Tab	Moves to previous frame
Alt+C	Moves focus to Contents tab (pressing Enter activates the Contents tab)

Key or Key Combination	Description
Alt+R	Moves focus to Search Results tab (pressing Enter activates the Search tab)
Right Arrow/Left Arrow	Moves between tabs
Select a view tab, press Enter	Switches views
Select a view tab, Up Arrow	Switches and moves to a view
Alt+S	Moves to the Search Entry field
Ctrl+P	Prints the current page or active frame
Ctrl+F	Finds a string in the current page or active frame (when using embedded Help browser on Windows system or Internet Explorer)
Alt+Underlined Letter	Accesses a control associated with a label (most labels of controls have assigned mnemonics)

Change the Help Browser Settings

The help browser uses your operating system's settings for the font colors, styles, and sizes. Users with visual impairments can change some of these settings to increase the readability of the documentation.

In addition, on Windows platforms using Microsoft Internet Explorer, the help browser uses a component of Internet Explorer to display documentation. The changes you make to the display settings of Internet Explorer also affect the help display.

Follow these steps:

1. Open Microsoft Internet Explorer. Select Internet Options from the Tools menu.
The Internet Options dialog appears.
2. Click Colors, Fonts, or Accessibility to set the formatting options you want.
The Accessibility dialog appears.
3. (Optional) Specify a cascading style sheet (CSS) on the Accessibility dialog.
Click OK.
The Internet Options dialog closes.

4. Close the Internet Explorer.

The changes you made take effect immediately.

5. Restart the Diagram Trace Utility, open the Help perspective, and browse the documentation.

You should see the new font and color settings applied.

Note: For more information about creating a CSS, consult a CSS reference. The W3 Consortium (www.w3.org) has an extensive collection of information about CSS and links to valuable resources.

Index

A

- accelerator keys • 147
- accessibility
 - features • 16
 - key usage • 147
- accessing
 - Action Diagram, accessing • 105
 - controls • 148
 - menus • 147
- activating views • 30
- adding
 - line breakpoint • 96
 - method breakpoint • 101
 - PAD Code View editor • 69
 - Watch expression • 58
 - watch on a variable • 66
- annotations preferences, setting • 129
- application target environments • 11

B

- bookmarks
 - accessing Action Diagram • 105
 - adding • 105
 - removing • 106
- Bookmarks view • 37
- breakpoints
 - adding • 96
 - controlling program execution • 94
 - description • 15
 - disabling • 96
 - enabling • 96
 - removing • 97
 - setting conditions on • 100
 - setting hit counts on • 98
- Breakpoints view • 39
- browser settings, changing • 150

C

- changing
 - help browser settings • 150
 - main window layout • 107
 - variable value • 65
- closing views • 30
- code view regions, expanding and contracting • 16

- conditional breakpoints • 100
- configuring
 - debug session • 89
 - Diagram Trace Utility • 85
 - generated application • 83
 - TCP/IP • 145
- connection problems, troubleshooting • 145
- Console view • 43
- contacting technical support • 3
- current perspective • 22

D

- debug session
 - configuring • 89
 - debugging components • 13
 - exiting • 106
 - starting • 85
 - stopping • 85
- Debug view • 44
- default perspective • 22
- Diagram Trace Explorer view • 50
- Diagram Trace Utility
 - application connection • 93
 - customizing • 107
 - editor and view tabs appearance preferences • 119
 - editor folding preferences • 112
 - editor syntax coloring preferences • 114
 - exiting • 84
 - installing • 19
 - navigating • 21
 - overview • 10
 - PAD Code View editor preferences • 110
 - setting preferences • 108
 - starting • 29
 - step filtering preferences • 115
 - validating data • 65
- display orientation of Expressions view, setting • 55
- displaying variable type names • 56
- drop cursors • 32

E

- error log
 - clearing • 144
 - displaying error details in Notepad • 143

- exporting • 144
- importing • 144
- reviewing contents • 141
- viewing details • 142

Error Log view • 51

existing applications, issue • 12

exporting key mapping preferences • 132

exporting preferences • 75

expressions

- removing • 57
- valid combinations and examples • 14

Expressions view • 53

G

general key preferences, setting • 131

generated source file types • 72

H

hardware requirements • 19

help browser settings, changing • 150

hit counts • 98

I

importing

- error log • 144
- PAD files • 72
- preferences • 72

installing Diagram Trace Utility

- considerations • 19
- post-installation tasks • 20

issues for existing applications • 12

K

Key preferences, exporting • 132

L

line breakpoints

- adding • 96
- disabling • 96
- enabling • 96
- removing • 97

Logs

- Console view • 139
- Error Log view • 141

M

main menu bar

DiagramTrace menu • 26

File menu • 25

Help menu • 28

Navigate menu • 26

Run menu • 26

Window menu • 28

markers • 15

method breakpoint

- adding • 101
- disabling • 102
- enabling • 102
- removing • 103
- toggling • 61

modifying

- PAD view data • 13
- preferences • 107
- system variables • 13

N

navigating online help system • 149

netstat command • 89

NLS capability • 16

O

online help

- dynamic help support • 78
- F1 support • 78
- navigating • 149
- using • 77

opening views • 35

operator precedence, expressions • 13

Outline view • 60

P

PAD Code View editor • 67

- activating • 69
- adding • 69
- rearranging the order • 71
- setting preferences • 124
- tiling • 69

PAD statement

- executing • 10
- highlighting • 16

perspective

- current • 23
- default • 22
- reinstating default settings • 25

preferences

- exporting • 75
- importing • 72

R

- rearranging tabbed views • 32
- reinstating default perspective • 23

S

- setting
 - color preferences • 148
 - Diagram Trace preferences • 107
 - display orientation • 55
 - general preferences • 117
 - help system, navigating • 149
 - hit counts • 98
- show variable type names • 56
- software requirements • 19
- source file types, generated • 71
- source-level debugging • 10
- switch key combinations • 148

T

- tabbed views, rearranging • 32
- target environments • 11
- technical support, contacting • 3
- text editors
 - accessibility preferences • 128
 - annotations preferences • 129
 - general preferences • 126
- text wrapping in Expressions detail pane • 56
- toggling
 - method breakpoint in Outline view • 61
 - watchpoint in Outline view • 61
- traced program, execution • 94
- troubleshooting
 - configuration problems • 146
 - connection problems • 145

V

- variable type names, displaying • 56
- variable value, changing • 64
- Variables view • 62
- view
 - activating • 30
 - Bookmarks • 37
 - Breakpoints • 39
 - Console • 43
 - Debug • 44

- Diagram Trace Explorer • 50
- Error Log • 51
- Expressions • 53
 - opening • 35
- Outline • 60
 - rearranging • 30
- Variables • 62

W

- Watch expression
 - adding • 58
 - description • 14
 - disabling • 59
 - editing • 59
 - enabling • 59
 - toggling in Outline view • 61