CA InterTest™

Graphical User Interface Primer

Release 9.1.00

CA Technologies Product References

This document references the following CA products:

- CA InterTest® Batch
- CA InterTest® for CICS

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: Using the CA InterTest for CICS Eclipse User Interface 7

CA InterTest Main Window 8
- Toolbars 9
- Views 9

Online Help 14
- F1 Support 15
- Dynamic Help 15

Chapter 2: Before You Begin 17

Synchronized Processing 17

Chapter 3: Debugging a CICS or Batch Application 19

Connect to a Server 19
- CUI--Import CA Testing Tools Server Settings 19
- Add a New Server Connection 20
- Check the Server Status 22

Create a Project 23

Configure the Debug Settings 23
- Configure Debug Settings for CICS Applications 23
- Configure Debug Settings for Batch Applications 24

Select the Source Program to Monitor 26
- Demo Source Programs 28
- How to Use the Outline View 28
- View Monitoring Status 29

Start Program Execution 29

Evaluate and Correct Errors 30
- Determine the Cause of the Error 31
- Change the Value in TASKNUM for COBOL and PL/I Applications 32
- Change the Value in TASKNUM for Assembler Applications 32

Breakpoints 33
- When to Use Breakpoints 35
- Set Unconditional Breakpoints 36
- Change Breakpoints - CICS 38
- Change Breakpoints 40
- How to Remove Breakpoints 41

Bookmarks 42

Add Bookmarks 42
How to Remove Bookmarks 43
Resume Program Execution 43

Chapter 1: Using the CA InterTest for CICS Eclipse User Interface

CA InterTest for CICS provides an Eclipse-based graphical user interface (Eclipse UI) for debugging CICS and batch applications.

You can use the CA InterTest for CICS Eclipse UI to perform the following tasks:

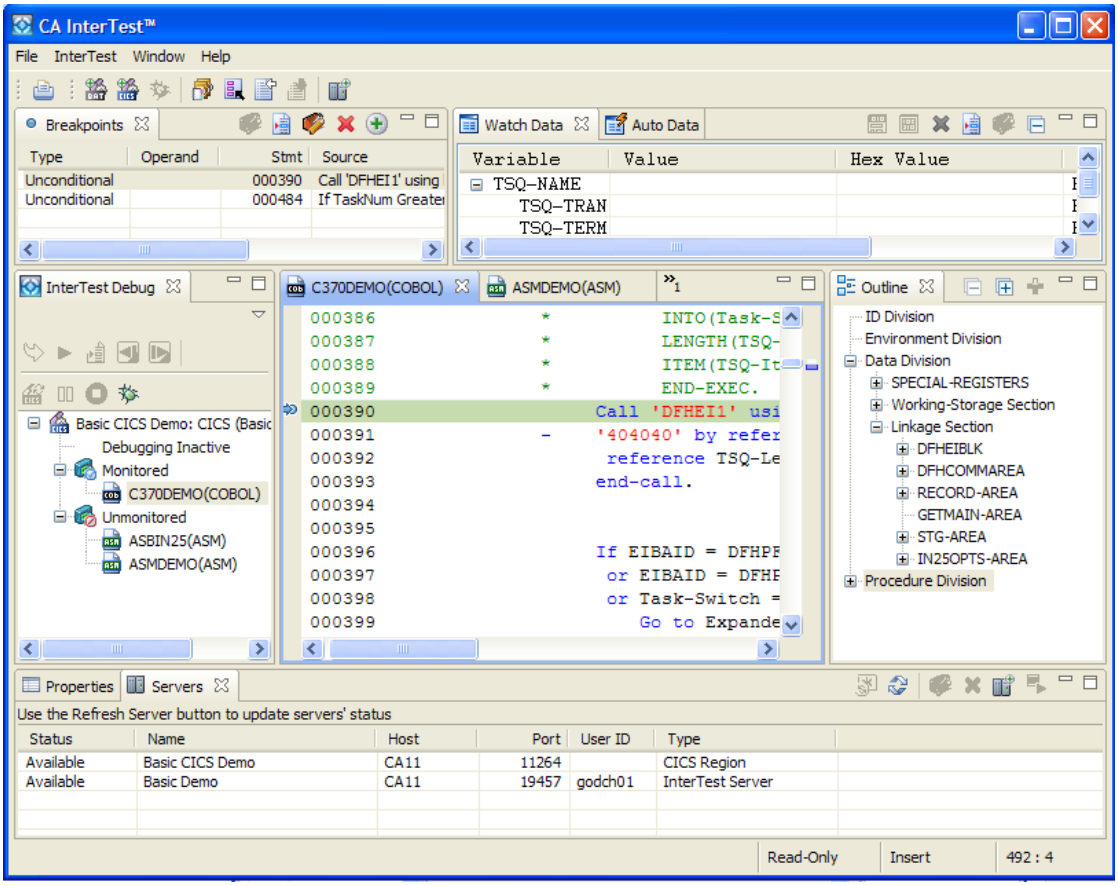
- Define projects for testing CICS or batch applications
- Import programs into your projects
- Set, remove, and modify breakpoints
- Pause, resume, and interrupt program execution
- View and modify data values of active programs
- Single-step forward through program statements
- Step backwards through previously executed statements
- Customize and personalize views

CA InterTest Main Window

CA InterTest contains a group of views that provide all the functions you need to debug your programs.

A *view* is a visual component within the CA InterTest window. It is typically used to navigate a hierarchy of information or display properties for a program listing, which displays the program being debugged. Modifications made in a view are generally done with dialogs.

The following illustration shows the CA InterTest main window with several views open.



Usually, only one instance of a particular type of view may exist within the CA InterTest window. Views might appear by themselves or stacked with other views in a tabbed notebook. To activate one of these views, click its tab. Only one view is active at one time. The active view is the one whose tab is highlighted. In this example, the program listing is active.

The exception to only one type of view existing within the CA InterTest window is the Program Listing area, where several imported program listing files can be open at one time. The listings behave like sub-views that are grouped together (or stacked) in the Program Listing area.

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

Toolbars

There are two kinds of toolbars in CA InterTest:

- Main toolbar
- View toolbar

The *main toolbar* is displayed at the top of the main window directly beneath the menu bar. The contents of this toolbar change based on the active view. You can find explanations of all the buttons on the toolbar in the online help.

There are also individual *view toolbars*, which appear in the title bar of a view. Actions in a view's toolbar apply only to the view in which they appear. The following example shows the Bookmarks view toolbar:

Note: Some view toolbars include a Menu button, shown as an inverted triangle that contain actions for the view.

Views

CA InterTest supports the following types of views:

Auto Data

Displays variables and their contents at a particular breakpoint.

Bookmarks

Displays all anchors (bookmarks) placed on a specific line of code.

Breakpoints

Displays all breakpoints (enabled and disabled) that are set in the vertical ruler of the Program Listing area.

Call Trace

Displays a list of the called subprograms in a load module based on the current breakpoint. This is a CICS only view.

Console

Displays read-only standard output messages from CA InterTest.

Error Log

Displays all warnings and errors generated by the common user interface. This is used by CA support and you can export this log when reporting an error.

Help

Displays the related help topics for the highlighted view.

InterTest Debug

Displays the monitored and unmonitored programs.

Outline

Displays an outline of the program that is currently open in the Program Listing area and lists the structural elements.

Properties

Displays a list of attributes, if any, related to another active view. Not all views have information to display in this view.

Register Data

Keeps track of general register values during a debugging session for Assembler programs.

Servers

Displays the servers that have been defined for CA InterTest to communicate with the mainframe.

Session Data

Keeps track of session-related information during a debugging session for CICS programs.

Statement Trace

Displays a trace of previously executed statements.

Tasks

Displays the tasks available for the currently active program.

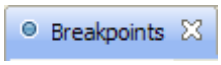
Watch Data

Keeps track of variables and other program-related information during a debugging session.

Use views to navigate a hierarchy of information (such as variable values), show breakpoints, bookmarks, console output, and various other activities.

CA InterTest saves any changes you make in a view across instances of the GUI.

Each view has a close button next to its name that lets you close the view. The following example displays the Breakpoints view tab with the close button.



Views also have their own toolbars and pop-up menus that contain actions you can perform that are specific to the view.

The following example displays the toolbar of the Breakpoints view with buttons for actions on the toolbar's right side:

Note: All views have minimize and maximize buttons on their toolbars.

You can perform the following actions with the buttons on the Bookmark's toolbar, as described in the following table:

Button	Description
	Configures the settings for the program name
	Shows the position in the program listing where the breakpoint is set
	Configures a selected breakpoint
	Deletes a selected breakpoint.
	Adds a global breakpoint
	Minimizes the view.
	Maximizes the view.

Open Views

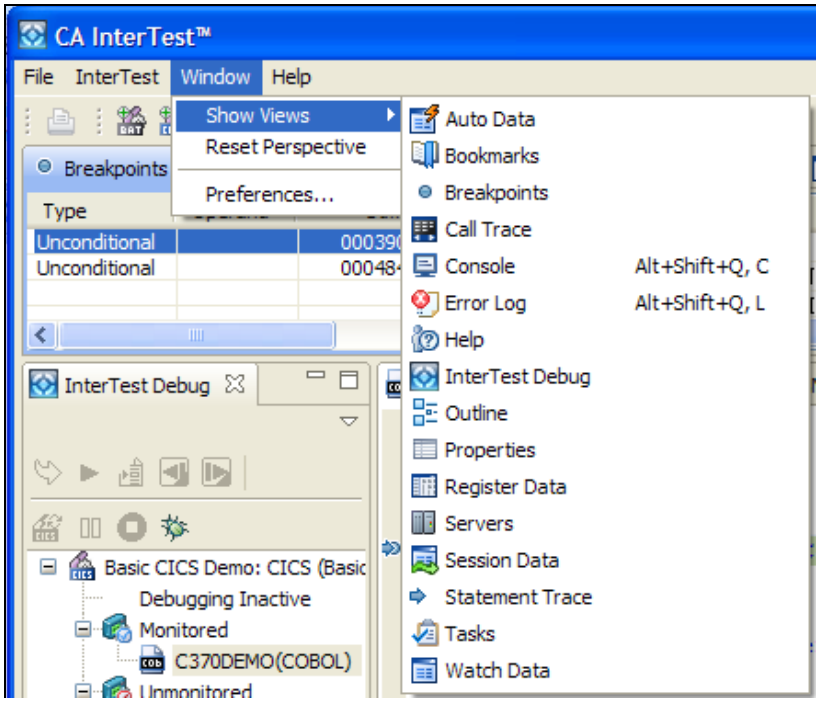
You can open a hidden view as needed. There are two ways of opening a view:

- Using the Window menu
- Using the shortcut keys

To open a view using the Window menu

1. Select Window, Show Views.

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



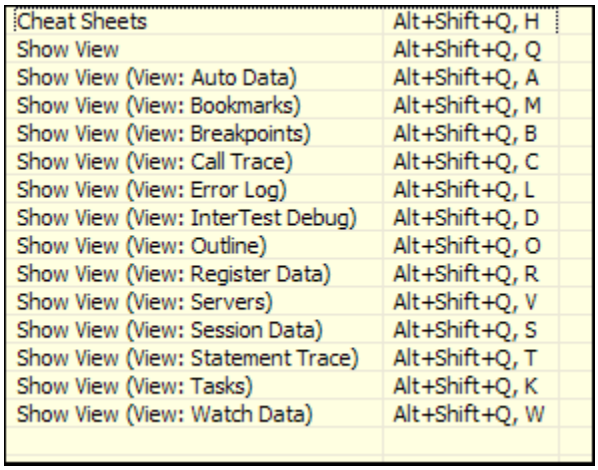
2. Click the view you want to open.

   The selected view is displayed in the CA InterTest main window.

To open a view using the shortcut keys

1. Press Alt+Shift+Q keys.

   A pop-up menu opens that lets you select and open the following views:



Cheat Sheets	Alt+Shift+Q, H
Show View	Alt+Shift+Q, Q
Show View (View: Auto Data)	Alt+Shift+Q, A
Show View (View: Bookmarks)	Alt+Shift+Q, M
Show View (View: Breakpoints)	Alt+Shift+Q, B
Show View (View: Call Trace)	Alt+Shift+Q, C
Show View (View: Error Log)	Alt+Shift+Q, L
Show View (View: InterTest Debug)	Alt+Shift+Q, D
Show View (View: Outline)	Alt+Shift+Q, O
Show View (View: Register Data)	Alt+Shift+Q, R
Show View (View: Servers)	Alt+Shift+Q, V
Show View (View: Session Data)	Alt+Shift+Q, S
Show View (View: Statement Trace)	Alt+Shift+Q, T
Show View (View: Tasks)	Alt+Shift+Q, K
Show View (View: Watch Data)	Alt+Shift+Q, W

2. Press the single letter for the view you want to see.

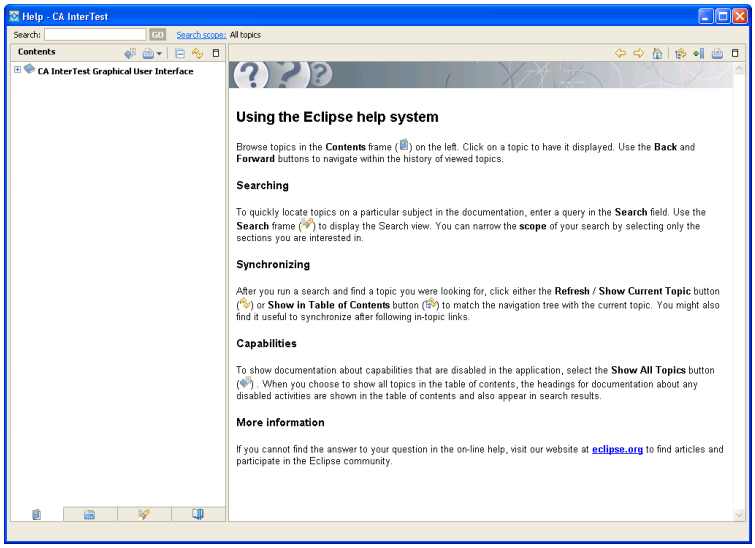
   For example, pressing the letter A will open the Auto Data view.

Online Help

CA InterTest provides extensive online documentation and context-sensitive help. You can request online help anytime while using CA InterTest 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 to the CA InterTest online documentation, which you can expand to display subsets of the documentation.



Search

Activates the CA InterTest online Help view and displays the Search panel in which you can enter text that you want to search for. Click Go to display all topics on the online documentation in which the search text appears.

Dynamic Help

Displays the help data as it is dynamically displayed to reflect the context of the view that is in focus. For example, when the focus is on Breakpoints, Dynamic Help will only display those topics relevant to breakpoints.

Note: The dynamic help feature is not supported in the Program Listing area pop-up menus, window controls such as menu items or buttons, or toolbar buttons.

Cheat Sheets

Opens pages to guide you through some of the processes. Each cheat sheet is designed to help you complete a specific task, and lists the sequence of steps required to help you achieve that goal.

Key Assist

Opens the Key Assist window where you can see shortcut key combinations.

Update InterTest

Opens the Install/Update dialog from which you will be able to install updates of the currently installed features, or new features.

About CA InterTest

Displays the About CA InterTest dialog, which provides information on the software release number, plug-in details, and configuration details.

F1 Support

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

If the control is located in the CA InterTest 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 is located on a dialog started by CA InterTest, 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.

Dynamic Help

When the CA InterTest Help view is active in CA InterTest as you make each view active, the help data displayed is dynamically updated to mirror the view that is in focus.

Chapter 2: Before You Begin

This document describes how to use the Eclipse UI to debug CICS and batch applications. Use the CA InterTest for CICS Eclipse UI and this document to walk through the process and procedures that follow.

This tutorial uses the default CA InterTest for CICS library names. See Demo Source Programs (see page 28) for the names of the demo programs that are used in this tutorial.

To perform the steps in this tutorial, you must know the following information. If you do not already know this information, or if the default demo program names were changed during installation, contact your CA InterTest for CICS administrator.

From your IBM installation

- User ID
- z/OS host name

From CA InterTest for CICS

- Path to the file containing Testing Tools server settings to be imported. If the application was installed properly, you have a set of servers that are properly configured for your site.
- CA InterTest for CICS PROTSYM data set names

For the CICS demo sessions, you also need the following information:

- CICS host name
- Path to the file containing CA InterTest CICS server settings to be imported

Synchronized Processing

CA InterTest for CICS searches the symbolic files for the programs that you specify to be monitored for testing and tries to match a symbolic file with the same date and time as that of the load module. If a match cannot be found, the Select Symbolic dialog opens. If this dialog appears, do one of the following:

- Select the version that you want the mainframe debugger to use and click Use.
- If you do not want to monitor the file, select Do Not Monitor.
- For batch applications, you can click Abend Stop to end the debugging session.

Use this feature to maintain synchronized processing at all times.

Chapter 3: Debugging a CICS or Batch Application

The following process describes how to use the CICS and batch basic demo sessions.

To debug a program using the Eclipse UI:

1. Connect to a server (see page 19).
2. Create a project (see page 23).
3. Configure the debug settings (see page 23).
4. Select the source program to monitor (see page 26).
5. Start the program execution (see page 29).
6. Evaluate and correct errors that the program identifies (see page 30).
7. Set and remove breakpoints (see page 33).
8. Add and remove bookmarks (see page 42).
9. Resume program execution (see page 43).

Connect to a Server

Before you begin performing tasks with CA InterTest for CICS, you must connect to a server. You can import server setting and you can add new server connections to the Eclipse UI.

CUI--Import CA Testing Tools Server Settings

Connect to a Testing Tools server before you start a debugging session. The GUI component runs on Windows and the debugging session runs on z/OS. This server provides the communication between the Eclipse UI and the mainframe platform.

You may need to import the Testing Tools server settings the first time you start the Eclipse UI component or whenever your CA InterTest for CICS administrator updates any of the server settings.

If the application was installed properly, you have a set of servers that are properly configured for your site.

Follow these steps:

1. Start CA InterTest.

   The CA InterTest main window opens.

2. Click File, Import.

   The Import dialog opens.

3. Click Next.

   The Import Server Settings dialog opens.

4. Type or browse for the file path to the file containing the server settings. Obtain this information from your CA InterTest for CICS administrator.

5. Click Finish.

   You are connected to the specified server.

Add a New Server Connection

If you have not already established a connection to the server that you want to use, you can add a server connection using the Eclipse UI. You must have the host and port details of the server to which you want to connect.

Follow these steps:

1. Select InterTest, Add New Server.

   The New Server Connection dialog opens.

2. Enter the details for the new server connection.

   The following field values are required to create a new server connection:

   Name

   Specifies the name of the server connection. You can use uppercase and lowercase alphanumeric characters, and embedded spaces when entering a value in this field.

   Note: For the purposes of this tutorial, specify the name "Basic Demo" in this field.

   Host

   Specifies the z/OS host name assigned by your installation. This is the host name used for TSO or CICS logon.

   Port

   Specifies the TCP/IP port number of the server. If there is a default port number displayed, verify it is correct. The port number is updated in the Preferences dialog.

   - For a CICS region—This field must be the port number configured in the CICS region.
   - For a Testing Tools server—This field must be the port number that is configured for the Testing Tools server running on z/OS UNIX System Services (USS).

   Use as Default Server

   Specifies that this server is selected when you access the Batch Link Queue or Schedule Table.

   CICS Region

   Defines a connection that is used when a CA InterTest for CICS for CICS debug session is initiated.

   Select this box to initiate a CICS debug session. This action displays the CICS Security Enabled check box. If you select this box, you can secure your connection by using a user ID and password.

   Clear the CICS Region check box to define a server connection.

Default: Off

(Optional) User ID

Specifies the user ID for logging on to a z/OS system. Enter your user ID in this field.

(Optional) Save Password

Specifies whether you want to save the password. If you save the password, CA InterTest for CICS preloads the saved password into the Logon dialog whenever you are prompted to enter your user ID and password. If you do not save it, you are prompted for your password whenever you log on to CA InterTest for CICS. Until the password is deliberately reset or a logon fails, you are prompted for a user ID and password only when you first use a server. Check this box to save your password.

3. Click Finish.

   The new server is created and added to the list of servers in the Servers view. The Servers view shows the details of the new server.

Check the Server Status

The Server view displays the status of the servers you are connected to so that you can see if the server is active.

Follow these steps:

1. Click Window, Show Views, Servers, on the CA InterTest main window toolbar if the Servers view is not already visible.

   The Servers view opens, displaying the name of the server and its status.

   The server has one of the following statuses:

   Available

   The GUI is connected to a mainframe server that is running.

   Not Available

   The GUI failed to connect to the server and is not running. The reason for the failure is noted next to the server status.

2. If the Status column is blank, click the Refresh Server button  on the Servers view toolbar.

   The system may take several seconds to respond.

3. If the status remains blank or Not Available, contact your CA InterTest administrator.

Create a Project

After you are connected to a CA Testing Tools server, create a project in which to debug your program.

Follow these steps:

1. Click the Create Project button  on the CA InterTest main window toolbar.

   The New CICS Project dialog opens or New Batch Project dialog opens, depending on the demo session.
2. Enter **Basic Demo** in the Project Name field.
3. Check the Use default location box if it is not already checked.
4. Select Basic Demo from the InterTest Server drop-down list.
5. For CICS applications, select Basic Demo from the CICS Region drop-down list.

   Note: Do not select Import files once project is created box.
6. Click Finish.

   The InterTest Debug view displays your project name, project type, the server name, the status indicator, and two empty folders: Monitored, and Unmonitored.

Configure the Debug Settings

Before you execute a program, set the debug settings for your project. The debug settings dialog is different for CICS and batch applications. This dialog is also displayed when you start the program execution.

Configure Debug Settings for CICS Applications

Follow these steps:

1. Right-click the project name on the InterTest Debug view, and select Configure Session Debug Settings.

   The InterTest Debug Settings dialog opens.
2. Ensure that the following fields are completed:

   InterTest Server

   Displays the Testing Tools server name.

   CICS Region

   Displays the CICS region name.

3. Click Finish.

   The debug settings are configured.

Configure Debug Settings for Batch Applications

Follow these steps:

1. Click the project name on the InterTest Debug view, right-click, and select Configure Session Debug Settings.

   The InterTest Debug Settings dialog opens with the main tab in view.

2. Click the InterTest Server drop-down list, and select Basic Demo.

3. Select the PROTSYM tab.

4. Click Add, and add CAI.PROTSYM as the PROTSYM data set name, or the data set name provided by your CA InterTest administrator, in the Add Dataset Name dialog. Click OK.

   The data set name appears in the PROTSYM Datasets area.

5. Click the JCL Settings tab.

   The InterTest Debug Settings dialog opens.

6. Complete the following fields:

   InterTest Server

   Displays Basic Demo. If Basic Demo is not showing, select it in the box.

   Conversion JCL Dataset(Member Name)

   Describes the mainframe data set name and member name. Type **CAI.CAMRJCL(DEMOJCL)** or a data set name provided by your CA InterTest administrator.

   Dataset Type

   Describes the type of data set. Select IBM PDS, or the data set type appropriate to the one provided by your CA InterTest administrator.

   Job Step Name

   Describes the name of the job step. Type **Step1**.

   PROCLIB List

   Leave empty.

   Converted/Submit JCL Dataset(Member Name)

   Describes the output data set where the converted JCL resides. Type **HLQ.MYJCL(BATCHDMO),** where HLQ is the high level qualifier you choose, or any allocated PDS.

7. Click the Convert JCL button.

   The converted JCL is placed in HLQ.MYJCL(BATCHDMO).

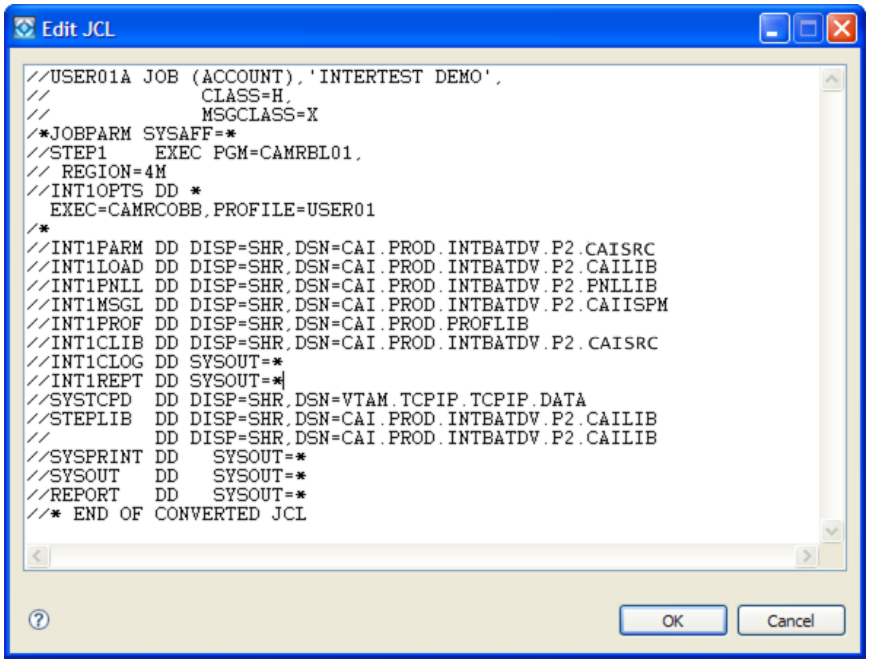
8. After the JCL conversion is complete, click the Edit JCL button to edit the converted JCL.

   The Edit JCL dialog opens.

9. Make the following changes to the Edit JCL dialog:

   - Change the job card to conform with your installation standards.
   - Change PROFILE=YOURID to PROFILE=(your TSO User ID).

   The following illustration shows the edited JCL:



10. Make sure the data set you are editing is not being used in TSO or CA Roscoe.
11. Click OK.

    The Edit JCL dialog closes and you return to the InterTest Debug Settings dialog.

12. Click Finish.

    You receive a message indicating that the file has been saved successfully. Click OK. You return to the InterTest main window.

Select the Source Program to Monitor

Now you are ready to look at the programs and instruct CA InterTest for CICS to monitor a program. As part of monitoring a program, the application automatically prevents all abends during execution.

Follow these steps:

1. Verify that the program to be debugged has been compiled and post-processed into the PROTSYM files.
2. Click the InterTest Debug view so that it is the active view.
3. Click the project name on the CA InterTest Debug view.

   The project name is highlighted.
4. Click the Import PROTSYM Files button  on the CA InterTest main window toolbar.

   The Import PROTSYM Files dialog opens.
5. Enter the following information:

   InterTest Server

   Defines the Testing Tools server. Select Basic Demo.

   PROTSYM Data Set

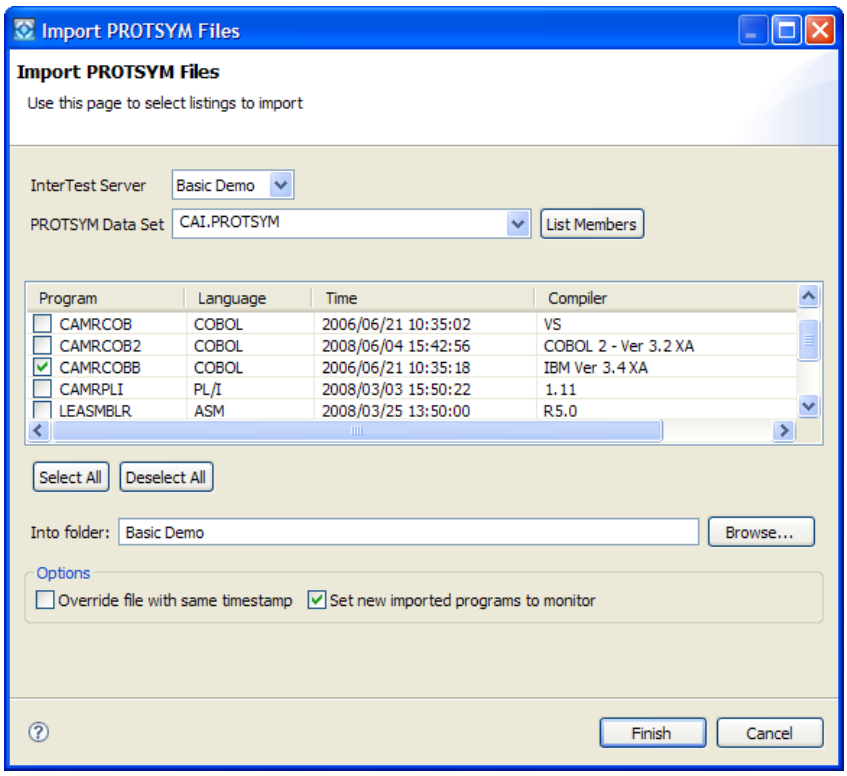
   Defines the fully qualified data set name of the PROTSYM VSAM file.

   Type **CAI.PROTSYM,** or substitute CAI.PROTSYM with the DSN provided by your CA InterTest for CICS administrator.

   If you receive an error about failure to retrieve a member list, check with your CA InterTest for CICS administrator.

   Note: You may be prompted to log on to your mainframe TSO here, if you have not already done so through CA InterTest for CICS.
6. Click List Members.

   The table is populated with the programs in the data set you chose.



7. Scroll down through the table until you reach the program that you want to use. For the names of the demo programs to use in this tutorial, see Demo Source Programs (see page 28). Select the program by checking the box next to it.

8. If the Into folder box is empty, click the Browse button, and select the Basic Demo folder to import the program into.

   The name Basic Demo appears in the Into folder box.

   Note: The CA InterTest for CICS administrator may have changed the name of the sample programs when installing the application. If you do not see the name of the demo program that you want to use, check with your CA InterTest for CICS administrator to find out the correct name.

9. Ensure that the two check boxes in the Options section are selected.

10. Click Finish.

    The demo program displays in the InterTest Debug view in the Monitored Programs folder.

11. Double-click the demo program name on the InterTest Debug view.

    The program listing displays in the Program Listing area, with an outline of the program in the Outline view.

Demo Source Programs

CA InterTest for CICS installs demo programs to use for this tutorial that are designed for use with the CA InterTest for CICS Eclipse UI tutorial. Use the compile procedures created during installation to compile a demo program, populate the PROTSYM, and link the program.

Note: The default names for the demo programs for each type of application may have been changed by your CA InterTest for CICS administrator during installation.

CICS Demo Source Programs

The CICS source programs for this tutorial are as follows:

Name of Program	Language Type	Transid
COBDEMO	Enterprise COBOL	DEMC
DB2DEMO	Enterprise COBOL	DEMD
PLIDEMO	PL/I	DEMP
ASMDEMO	Assembler	DEMA

Batch Demo Source Programs

The demo source programs for batch applications are as follows:

Type of Program	Name of Program
Assembler	CAMRASM
COBOL	CAMRCOBB
PL/I	CAMRPLI

How to Use the Outline View

The Outline view displays an outline of the program that is currently active in the Program Listing area and lists the structural elements. This makes it easier to display any section of your program. The contents of the Outline view are specific to the active program listing. The Outline view contains a top level node for each action.

- To expand a node, click the plus sign (+) next to the text.
- To collapse a node, click the minus sign (-) next to the text.

  Note: Selecting a variable or label name repositions the program listing to the line containing the name. If the location is collapsed, the area is automatically expanded.

View Monitoring Status

The InterTest Debug view lists the programs in a tree that identifies them as monitored or unmonitored. You can view the programs under these categories by clicking the plus (+) to expand the Monitored and Unmonitored nodes.

Start Program Execution

The process to start the program execution is different for batch and CICS applications. Follow one of these procedures to begin debugging your program.

For batch applications

Follow these steps:

1. Select the program that you want to monitor under the Monitored node in the Debug view.
2. Click the green bug button  on the toolbar.

   The batch job is submitted and the initial intercept is displayed.
3. Click the green arrow button to allow the program to run.

   The program will run until it encounters an abend.

For CICS applications

Follow these steps:

1. Under the Monitored node on the InterTest Debug view, select the program that you want to monitor and click the green bug button  on the toolbar.

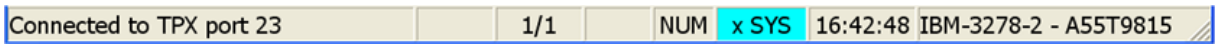
   The InterTest Debug Settings dialog opens.
2. Ensure that the debug settings are defined as you want them. For more information about configuring the debug settings, see Configure the Debug Settings (see page 23).
3. Click Connect.

   CA InterTest notifies CA InterTest for CICS that your program is ready for debugging.
4. Go to the mainframe CICS display (green screen), and type in the transaction identifier of the demo program as listed in Demo Source Programs (see page 28).
5. Press Enter.

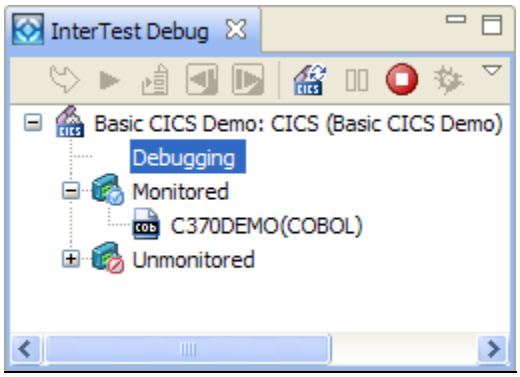
   The Welcome screen appears.

6. Press Enter on this CICS mainframe screen.

   The demo program begins execution. Your CICS session will indicate that it is in xSystem state:



7. Return to the CA InterTest for CICS Eclipse UI to continue with your debugging session. Notice that the InterTest Debug view now displays the debug status as Debugging.



Evaluate and Correct Errors

When a program encounters an error during execution, and abend occurs and stops the program.

At the abend, the following events occur:

- The appropriate program listing causing the error is activated (if the program was not displayed earlier).
- The program positions itself at the statement causing the error.

  When the program stops, we say that it halts the program at a breakpoint. This is done automatically when an abend occurs, or the application halts a program at a breakpoint that you have previously set.

When an abend occurs:

- A small blue arrow appears in the ruler to the left of the line causing the error, and the line is highlighted.
- The status in the InterTest Debug view now displays a diagnostic message. This message explains that the error was caused by improperly formatted data.
- In this tutorial, the demo program detects an error at the ADD instruction. Execution of that ADD instruction triggers an abend.

Once a program is stopped, you can use the CA InterTest for CICS testing and debugging facilities to do the following tasks. When debugging your own programs, you will typically perform one or more of these activities.

- Examine the source listing.
- Examine and modify main and auxiliary storage (memory) to detect and correct errors.
- Set and remove breakpoints and bookmarks.
- Examine a program's statement trace.
- Keep data items in a watch view to observe changes in their values.
- Abend your task with or without a dump.
- Go around a problem by resuming program execution from a location other than the one at which the program is currently stopped.
- Execute the program in single-step mode; that is, the program executes one verb and then stops.

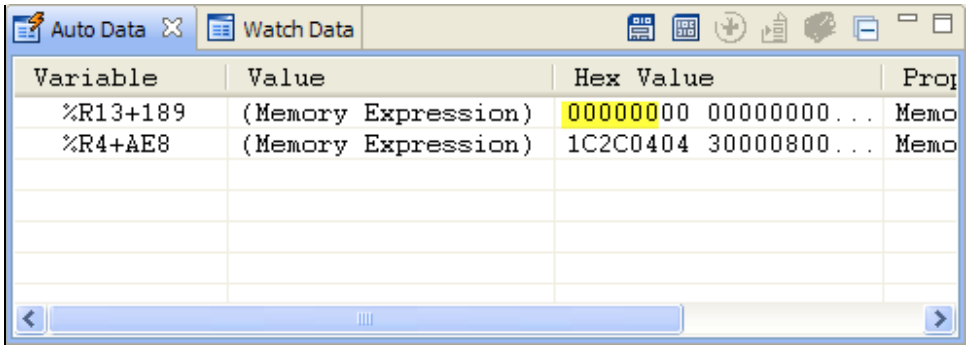
Determine the Cause of the Error

When an abend occurs, identify the cause of the error and correct it before resuming the program execution. Use the Auto Data view to see the values of the variables that are associated with a statement. Use the Watch Data view to see the value of data items that you want to watch.

Example:

In this example, an Assembler program has encountered an error at the TASKNUM statement. The value of TASKNUM is displayed as an expression (%R13+189) in the Auto Data view. The left column shows the base + displacement for the name of the data item. In the columns to the right are the field's display value. Assembler programs show as Memory Expression, and hex value at that offset.

e



Look at the contents of TASKNUM. Confirm that the value stored in TASKNUM is not a valid packed decimal value by examining its current value in the Auto Data view; that is, its value prior to the execution of the AP statement that triggered the ASRA. Instead, it contains low-values (binary zeros – the three highlighted bytes). Assembler does not allow you to add a packed decimal value (=P'1') to a field initialized with low values.

Change the Value in TASKNUM for COBOL and PL/I Applications

You have identified and confirmed the cause of the problem. Now you want to fix it.

Follow these steps:

1. Right-click the variable TASKNUM in the Auto Data view. Click Change Value.

   The Data View Value dialog opens.

2. Select the Display value button.

3. Change the value of TASKNUM displayed in the box to a numeric zero (type a numeric zero).

   Note: You do not have to know the type of data (binary, packed, and so on) or the length of TASKNUM. You can change the contents of a field by overtyping the desired bytes.

4. Click OK.

   TASKNUM now contains a packed decimal zero.

Change the Value in TASKNUM for Assembler Applications

Follow these steps:

1. Right-click one of the following variables in the Auto Data view

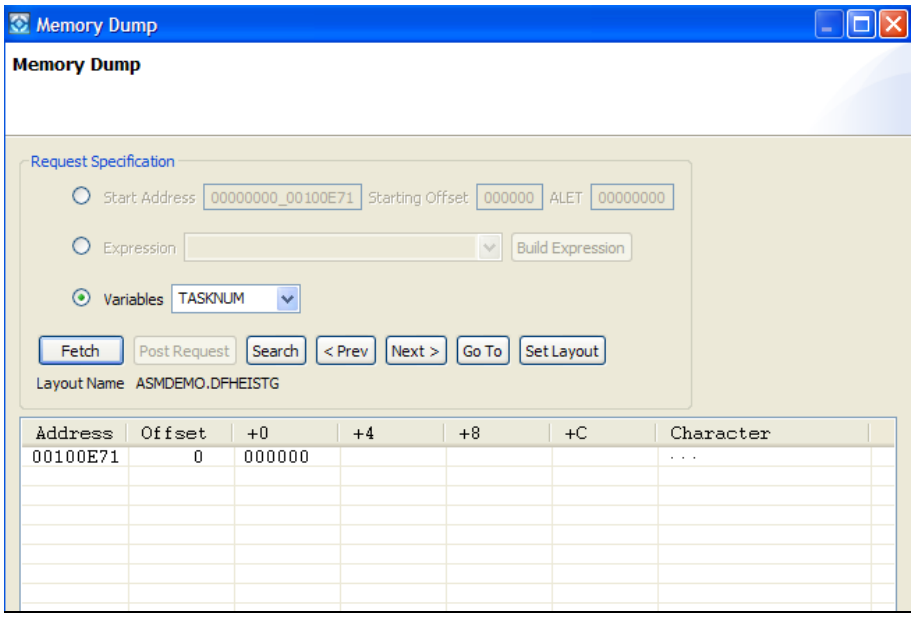
   - %R13+189–for CICS applications
   - %R13+634–for batch applications

2. Click Memory Dump.

   The Memory Dump dialog opens and displays the memory address for TASKNUM in the Address column. To the right under offset +0 the hex value displays, to the far right is the character value. The first three bytes (highlighted yellow) are the value of TASKNUM.

3. Click the Variables button, type TASKNUM, and click the Fetch button.

   The Memory Dump dialog opens, and the three bytes of TASKNUM display.



4. Overtype 000000 with 00000C, and press Enter.
5. Click Close.

   The Auto Data view now displays TASKNUM with a packed decimal zero.

Breakpoints

A *breakpoint* is an intentional stopping or pausing place in a program, put in place for debugging purposes.

When you open or activate a program listing in the Program Listing area, the breakpoints added to that program appear as blue circles. When the mouse hovers over a blue circle, the following details of a breakpoint are displayed:

- Breakpoint type
- Program name
- Statement number (in square brackets)
- Source code text

There are six types of breakpoints.

Automatic

The program stops because CA InterTest for CICS detected and prevented an error. When a program is stopped at an automatic breakpoint, you can either correct the error or go around it. You set all other breakpoints.

Conditional

The program stops at the location you specify if a condition is met. Optionally, conditional breakpoints can be set to stop at *any* instruction if a condition is met.

Request

The program stops at every CICS command or macro, at certain CICS commands or macros, or at calls to PL/I, DB2, or software.

Single-step

The program stops after executing one or more verbs.

Unconditional

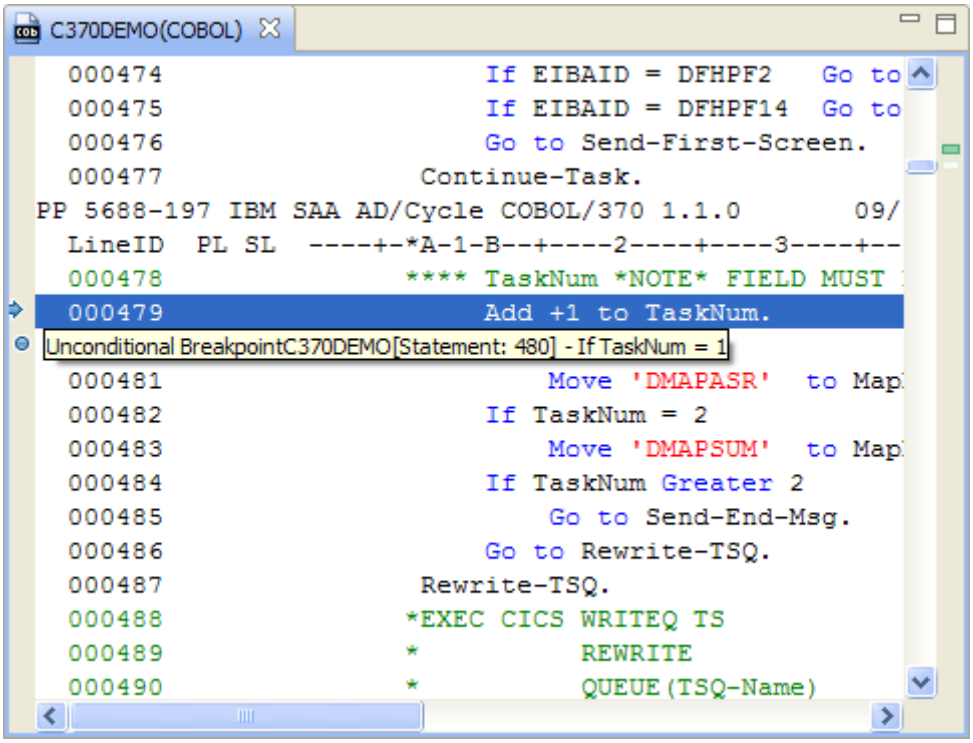
The program stops at the location you specify, just before the statement is executed.

Variable Change

The program stops at any location if the value of a specified variable has changed. This is a special type of conditional breakpoint.

Example:

In this example, execution stops just before the statement Add +1 to TaskNum is executed. A blue arrow points to that statement. This illustration displays the information when the mouse hovers over a blue circle.



When to Use Breakpoints

You can set breakpoints at any time—before you execute the program and when the program is stopped. Although where you decide to set breakpoints depends on the specifics of your program, you might want to set breakpoints in the following places:

- At the beginning of the Procedure Division, so when the program executes you can set additional breakpoints
- At paragraph names, so you can examine the contents of variables at the start of sections
- Before a call, so you can dynamically control the program path
- At each location named in an EXEC CICS HANDLE CONDITION, so you can verify error handling

Set Unconditional Breakpoints

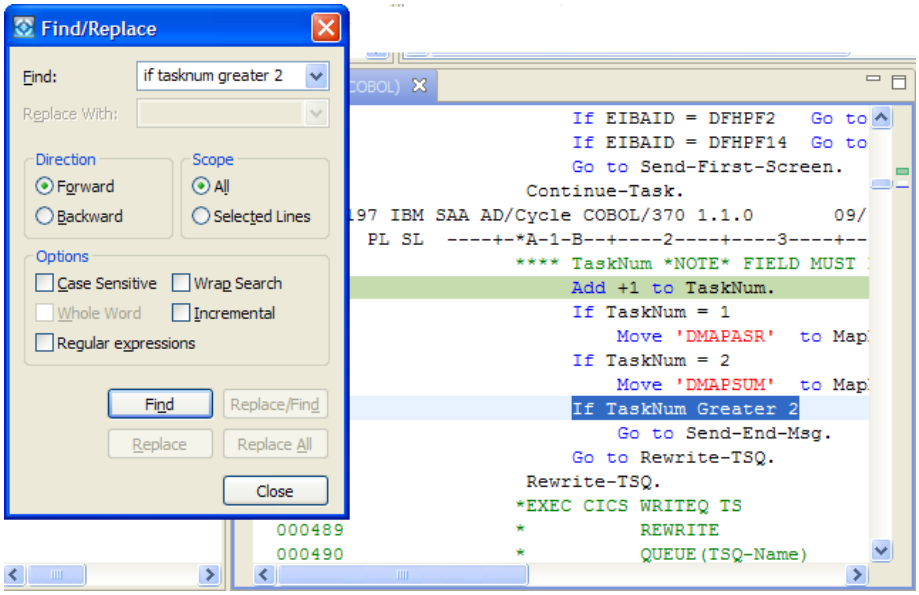
At an unconditional breakpoint, the program stops at the location you specify, just before the statement is executed. Let us see how to set a breakpoint at the following IF statement. You set unconditional breakpoints directly on the Program Listing area.

Follow these steps:

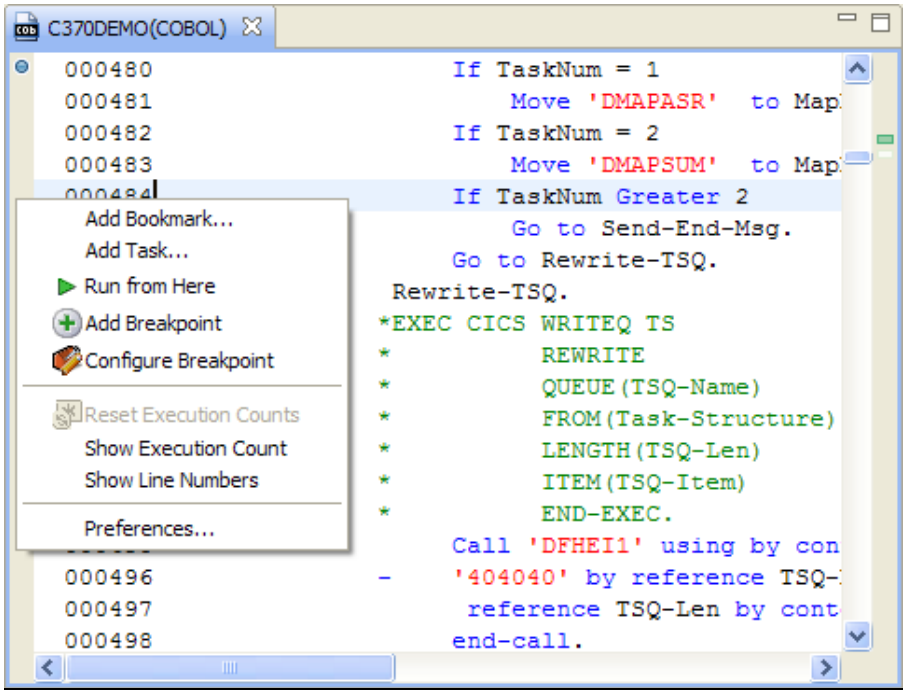
1. Click the demo COBOL program listing on the Program Listing area, so its name is highlighted in its tab.
2. Press <Ctrl> + F on your keyboard.

   The Find/Replace dialog opens.
3. Type **If Tasknum Greater 2** in the Find box, and click Find.

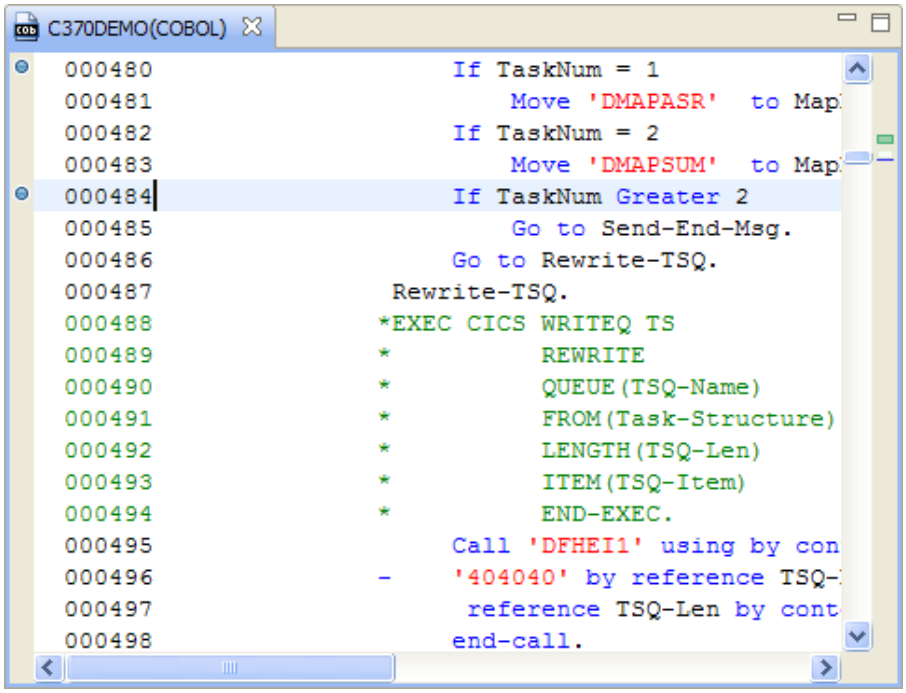
   The demo program positions itself so that the line you searched for appears in the program listing.



4. Right-click the ruler to the left of the line you have selected (statement 484), and select Add Breakpoint.



A blue circle appears to the left of the ruler, and a blue mark appears in the right-side vertical ruler. By default, the Add Breakpoint dialog adds an unconditional breakpoint if the listing line is within the Procedure Division, and a variable change breakpoint if the line is within the Data Division.



Note: You cannot add a breakpoint in a comment line or outside the source code section.

Change Breakpoints - CICS

The process for changing breakpoints is different for CICS and batch applications. The following procedure describes how to change a breakpoint in a CICS application.

Follow these steps:

1. Right-click the blue circle to the left of the line that contains the unconditional breakpoint that you want to change, or double-click the breakpoint in the Breakpoints view, and select Configure Breakpoint in the dialog.

   The Update Line Breakpoint dialog opens.

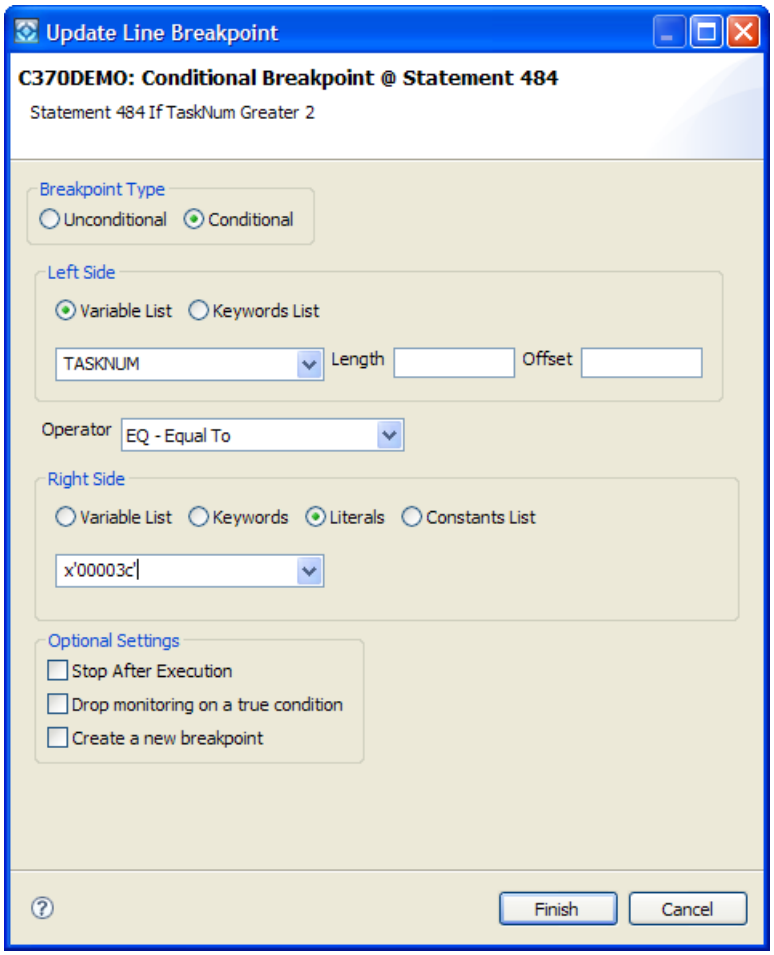
2. Select Conditional breakpoint under Breakpoint Type.

   The other fields become active.

3. Click the Variable List button, and select TASKNUM from the Variable List.

4. Click the Literals button and change the literal value to the new value where you want the breakpoint.

   The following illustration displays the finished dialog showing a conditional variable at TASKNUM=x'00003c'.



5. Click Finish.

   The dialog closes and the breakpoint displays in the Breakpoints view as a conditional breakpoint.

6. Repeat steps 1 through 5, and make statements 480 and 482 unconditional breakpoints.

7. Change the breakpoint at statement 484 back to an unconditional breakpoint.

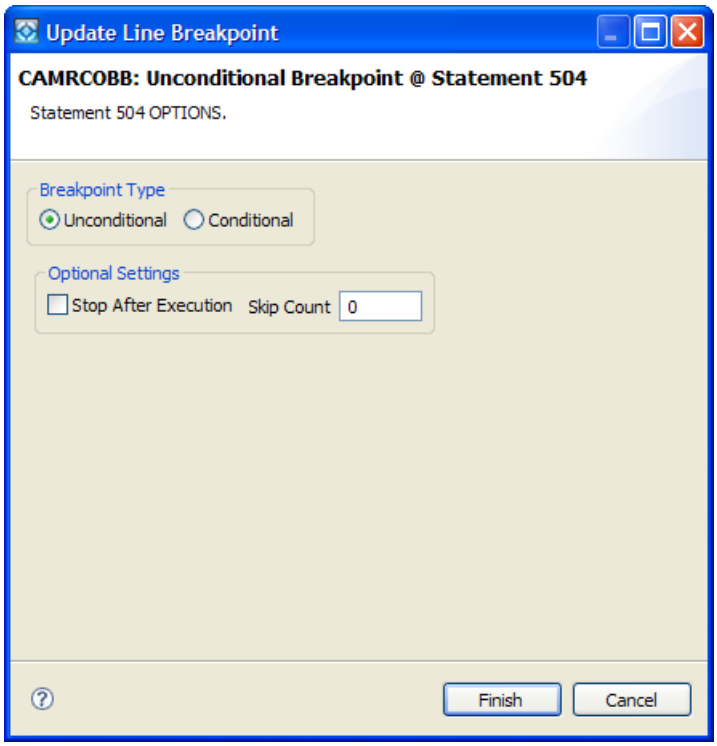
Change Breakpoints

The following procedure describes how to change breakpoints in batch applications.

Follow these steps:

1. Right-click the blue circle to the left of the line where you have set an unconditional breakpoint, and select Configure Breakpoint in the dialog.

   The Update Line Breakpoint dialog opens.



2. Click the Conditional breakpoint option button.
3. Change the fields as follows:

   Left Side–LOOP-OUT

   Operator–GT - Greater Than

   Right Side–x'00'

   Note: Character literals do not need to be prefixed with a C. The system automatically recognizes them as characters. Thus, 2 is treated as F2, whereas hex literals must be prefixed with an X.

4. Click Finish.

   The dialog closes and the breakpoint displays in the Breakpoints view as a conditional breakpoint.

How to Remove Breakpoints

As you continue testing and debugging it is good practice to delete breakpoints you no longer need, so that the program will not stop unnecessarily.

It is easier to delete individual or multiple breakpoints from the Breakpoints view when you have a number set throughout a large program and you do not want to search through the source listing for them.

You can delete breakpoints in the following ways.

- Hover the mouse on the vertical ruler to the left of the statement whose breakpoint you want to delete, right-click, and select Delete Breakpoint. The blue circle disappears, and there is no longer a breakpoint at this line.
- Select the breakpoint from the list of breakpoints in the Breakpoints view, right-click, and select Delete Breakpoint. You can also use the <Del> (Delete) key. Do not delete the unconditional breakpoint set at statement 484.

You can use the Eclipse UI delete several breakpoints at one time, follow these steps:

1. Click the first breakpoint on the Breakpoints view, hold down the Shift key, then click the last breakpoint you want to delete.

   The breakpoints you select are highlighted.

2. Right-click and select Delete Breakpoint.

   The breakpoints you select are deleted from the Breakpoints view and the program listing.

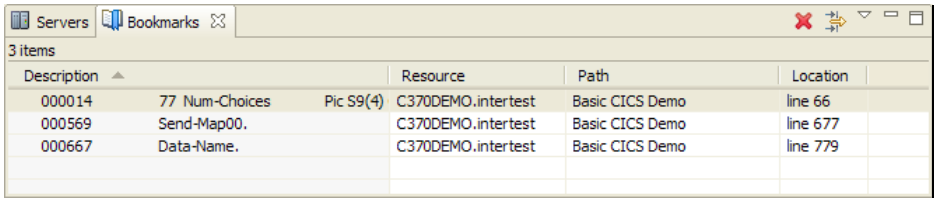
Bookmarks

Bookmarks help you to navigate quickly in the source code by marking individual locations. You can add and delete bookmarks as needed.

The Bookmarks view displays all bookmarks placed on a specific line of the program listing. The bookmarks appear in a table format that provides a description of each bookmark, the name of the program where the bookmarks are located, folder (path), and line number (location) in the Program Listing area.

Example

The following illustration displays a sample Bookmarks view, showing three bookmarks in the C370DEMO program:



Note: This view is not displayed unless you open the view by selecting Window, Show Views, Bookmarks on the CA InterTest main menu.

Add Bookmarks

You can add bookmarks to your program to help you navigate quickly through the program.

Follow these steps:

1. Double-click the demo program on the InterTest Debug view.

   The program listing displays in the Program Listing area.

2. Select the statement where you want to insert the bookmark.

   The listing is positioned at the line which includes the selected item.

3. Right-click the shaded vertical ruler on the left side of the listing view, and select Add Bookmark.

   The Add Bookmark dialog opens, with the contents of the listing statement displayed in the Enter Bookmark Name field.

4. Edit the name (description) for the bookmark and click OK.

   A new bookmark is set at the selected statement. This is indicated by a small rectangle  appearing next to the scroll bar. The Bookmarks view displays the details of the new bookmark.

In the Bookmarks view, after you add a bookmark, you can double-click the bookmark to open the program listing, if not already opened. The listing is positioned at the statement specified in the bookmark and highlights the statement.

How to Remove Bookmarks

Once you have created and used bookmarks, it is a good idea to delete them, so you do not clutter up your program or the Bookmarks view.

Select the line of code that contains a bookmark, and perform one of the following steps:

- From the program listing left-hand ruler, right-click your mouse, and select Remove Bookmark from the pop-up menu.
- From the Bookmarks view, highlight the bookmark and click the Delete button (the red X) on the Bookmarks view toolbar, or use the Delete key on your keyboard.

The bookmark is removed from the Bookmarks view and the program.

Resume Program Execution

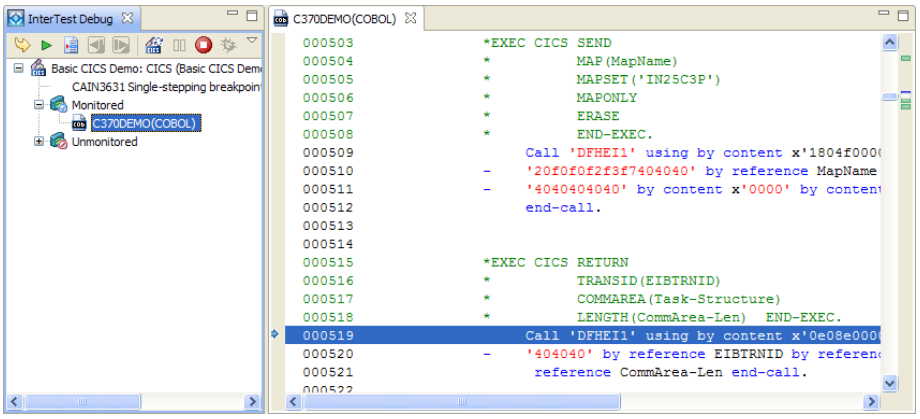
Resume the program execution after evaluating and resolving an error that the program identified or after the program has stopped at a breakpoint.

Follow these steps:

1. Click the Step Once button  on the InterTest Debug view toolbar to continue testing by resuming program execution.

   Program execution stops at the next statement. The debug status in the InterTest Debug view indicates that the program is at a single-step breakpoint and that shows the value that you updated in resolving the previous abend.

The following example shows the results of this step for the CICS COBOL demo.



2. Click the Resume Debugging button  on the InterTest Debug view toolbar.

   The demo program halts at the unconditional breakpoint you set previously in this tutorial. The InterTest Debug view window updates to reflect the reason for the stop

3. Click the Resume Debugging button again.

   The program continues until no further abends, breakpoints, or bookmarks are encountered.

4. When the program completes, do one of the following actions to end the debugging session.

   - For batch applications, the demo program displays the Debugging dialog to indicate that the program is complete. Click OK to terminate the debugging session. The tutorial is finished.
   - For CICS applications, the mainframe CICS screen displays the END DEMO SESSION screen. Press Clear or Enter on the mainframe CICS screen to end the CICS debugging session and then complete the next two steps.

5. In the Eclipse UI, click the Cancel Debug Session button  on the InterTest Debug view toolbar.

   The Cancel Debug Session dialog opens.

6. Click the Disconnect and Remove Breakpoints button, and then click Finish to terminate the connection with the server.

   The debugging status now shows Debugging Inactive in the InterTest Debug view.