

WHEN YOU NEED TO BE SURE

SGS

CyFlex® Test Run-Time Controller User Guide

Version 3

April 15, 2020

Developed by SGS North America, Inc.

Version History

Version	Date	Revision Description
1	6/26/2015	Initial publication
2	8/23/2018	Format with SGS brand
3	4/15/2020	Retrofit to new template Reflowed topic sequence in <i>Section 3 Starting and Exiting the TRC</i>

Document Conventions

This document uses the following typographic and syntax conventions.

- Commands, command options, file names or any user-entered input appear in Courier type. Variables appear in Courier italic type.
Example: Select the `cmdapp-relVersion-buildVersion.zip` file....
- User interface elements, such as field names, button names, menus, menu commands, and items in clickable dropdown lists, appear in Arial bold type.
Example: **Type**: Click **Select Type** to display drop-down menu options.
- Cross-references are designated in Arial italics.
Example: Refer to *Figure 1*...
- Click intra-document cross-references and page references to display the stated destination.
Example: Refer to *Section 1 Overview* on page 1.
The clickable cross-references in the preceding example are *1*, *Overview*, and on page 1.

CyFlex Documentation

CyFlex documentation is available at <https://cyflex.com/>. View **Help & Docs** topics or use the **Search** facility to find topics of interest.

Table of Contents

1	OVERVIEW	1
1.1	INTRODUCTION TO TRC	1
2	INSTALLING THE TRC	3
3	STARTING AND EXITING THE TRC	4
3.1	OPENING THE TRC GUI	4
3.2	LOGGING OUT AND HANDING OFF	5
3.3	REMOTE SUPPORT	5
3.4	EXITING THE TRC	5
4	NAVIGATING THE TRC WINDOW	6
5	LOADING A TEST REQUEST	12
6	EXECUTING A TEST REQUEST	15
6.1	STARTING OR RESTARTING A TEST REQUEST	15
6.2	SUSPENDING AFTER THE CURRENT ACTIVITY	16
6.3	TERMINATING AFTER THE CURRENT ACTIVITY	16
6.4	HOLDING THE CURRENT ACTIVITY	17
6.5	ABORTING THE CURRENT ACTIVITY	17
7	CLOSING A TEST REQUEST	18

List of Figures

FIGURE 1: TEST RUN-TIME CONTROLLER (TRC) WINDOW	4
FIGURE 2: ENTER LOGIN CREDENTIALS	4
FIGURE 3: ANNOTATED TRC WINDOW.....	6
FIGURE 4: TEST REQUEST SELECT/SEARCH WINDOW.....	12
FIGURE 5: RETURNED SEARCH RESULTS.....	13
FIGURE 6: RETURNED TEST REQUEST REVISION LIST	14
FIGURE 7: SELECT A TEST REQUEST ACTIVITY.....	15
FIGURE 8: SELECT STARTING POINT WINDOW	15

LIST OF TABLES

TABLE 1: TRC WINDOW COMPONENTS AND FUNCTIONS	6
TABLE 2: TEST REQUEST SELECT/SEARCH FIELDS	12

1 Overview

This document describes how to use the Test Run-time Controller (TRC), including:

- Installing TRC
- Starting and exiting the program
- Navigating the TRC Window
- Loading a test request
- Executing a Test Request
- Closing a test request

1.1 Introduction to TRC

The Test Request Suite (TRS) of software tools allows a test engineer to request specific tests on a unit under test (for example, an engine). An operator can execute the tests at a test bed (for example, test cell). The Test Run-time Controller is one application in the suite.

Test Run-time Controller (TRC) is the part of TRS that enables operators to execute the test request. It is integrated with CyFlex, the test system software that is used extensively for engine and component testing at Cummins. All components of TRS, including TRC, are implemented in the Java programming language for portability across operating systems.

Note:

The terms *unit under test* and *test bed* appear in this manual because the TRS is not limited to testing engines in test cells, but can also test components (turbos, after-treatment, fuel pumps) in test stands and test rigs.

TRC works with another application in the Test Request Suite called the Test Request Interface (TRI). This program enables engineers to create test requests. While TRI is executed primarily on Cummins Power Sweep Windows computers, it also works on CyFlex/Linux systems in test beds. Using TRI, operators can view test request specifications and fix any errors before executing the request.

Important:

TRC users must have a general understanding of TRI also. The TRC is described in a separate user manual as described in *Section 1.2 Supporting Documents* on page 2.

Engineers using the TRI create the test requests which store information in a combination of database and file-share. The TRS database (common to all of Cummins) keeps a short list of top-level information about the test requests. Use of this single database ensures that the test request name (a key identifier of the test request) is not duplicated within Cummins for a completely different test request.

A test request may evolve over time to include additional tests or fix errors in the specifications. Changes are identified by a revision number within the test request name. Revisions are identified by two numbers separated by decimal point (for example: 2.4). The first number is the major revision; the second number is the minor revision within the major revision. When TRI is used at the test bed for fixing errors, this typically changes the minor revision.

The specifications of the test request as well as attachments, that is, files referenced by the test request, such as `ecfg`, `cal`, and graphics files, are kept on a file-share accessible to the test cell and physically close for faster network access. By design, TRS keeps the files containing the test request name / revision combination free from subsequent changes and creates a new copy even if the revision number changes. This means the test request name / revision is viewable even years later.

Test requests are organized as a collection of activities. Examples include:

- Steady state mapping
- Execution of transient emission cycles
- Information and instructions to the operator
- Downloading of calibrations to the ECM

The Test Run-time Controller provides intuitive and flexible controls, letting the operator view all activities in a test request, and execute some or all of them.

TRC does not execute the activities but controls them through CyFlex. The result is the same as performing the activities manually from the CyFlex command line or using scripts. However, using TRC provides:

- Higher levels of standardization and automation to minimize transcription / omission errors from paper instructions
- More capability for unattended operation
- Better tracking of the status and history of activities in each test request

Note:

Suggestions for improvement to TRC are welcome. Communicate with CyFlex support personnel who can then convey suggestions to the TRS development staff.

2 Installing the TRC

The Test Run-time Controller (TRC) is installed on CyFlex test systems as part of a full installation of the Test Request Suite (TRS). CyFlex support personnel perform the installation.

Determine whether TRC is installed on a CyFlex system. Click the menu path **Menu – Favorites – Test Run-time Controller** to check whether the TRC icon exists.

3 Starting and Exiting the TRC

The Test Run-time Controller utilizes a Graphical User Interface (GUI). The actual task of executing test requests is done by a separate program in CyFlex which automatically starts with CyFlex. The GUI views what the test request is doing and gives the operator control.

3.1 Opening the TRC GUI

Execute the following steps to open the TRC GUI:

1. From the task bar at the bottom of the CyFlex screen, select the following sequence on the CyFlex menu:

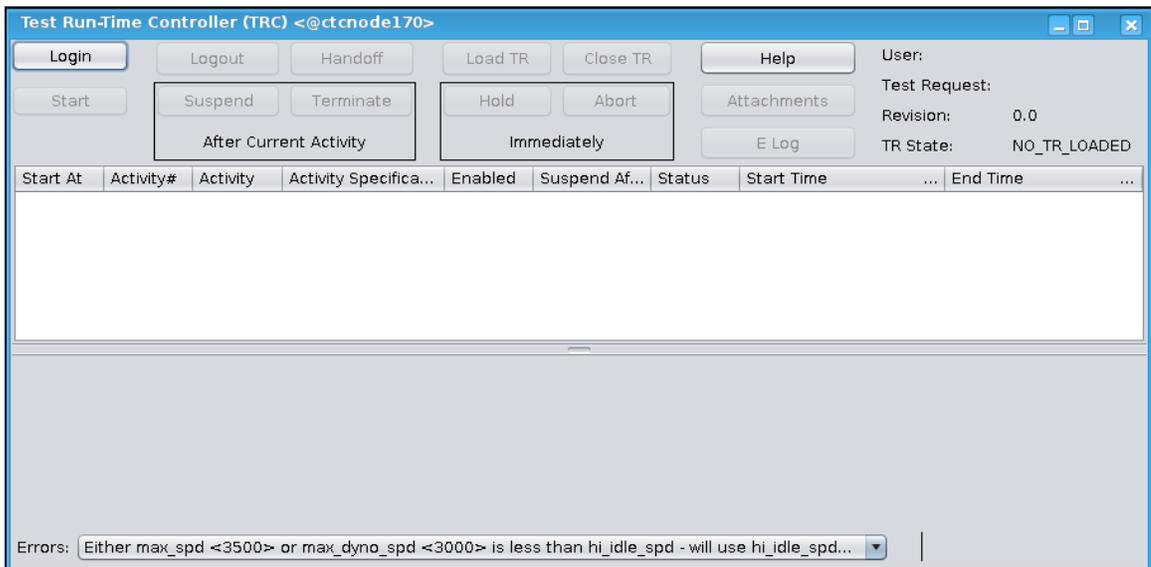
Menu - Favorites - Test Run-time Controller (TRC)

@Note:

Alternatively, open a terminal window and enter the command: `trc`

The Test Run-Time Controller (TRC) window is displayed as in *Figure 1*.

Figure 1: Test Run-Time Controller (TRC) Window



The **Login** and **Help** tabs are the only active selections.

2. Click the **Login** tab. The screen as in *Figure 2* is displayed.

Figure 2: Enter Login Credentials



Enter **Username** and **Password**.

If the CyFlex test cells are not migrated to Lightweight Directory Access Protocol (LDAP) authentication, follow these steps:

- a. In the *Login name* field, enter the test cell name in this format: `tcNNN` where *NNN* is the test cell number.
For example: `tc214`
- b. Enter at least one character in the **Password** field.
- c. Click **Login**.

3.2 Logging Out and Handing Off

The TRC GUI does not actually execute the test; programs in CyFlex do. This allows the operator to log out of the TRC GUI or hand off responsibility for managing the test request to someone at the end of a work shift for example, without disrupting a test request in progress. This is done using the control buttons described in *Section 4 Navigating the TRC Window* on page 6.

The **TRC** window displays the status and progress of a test request if one is loaded. An operator does not have to be logged in for this but must login to control the test.

3.3 Remote Support

If test request does not work properly, there are two ways in which CyFlex support personnel can provide support remotely.

1. One is to use VNC, which gives remote access to the CyFlex desktop, including TRC. Whatever the operator or support person does on the desktop is visible to the other.
2. An alternate method when VNC is not viable is for support personnel to connect to the test cell computer, open a parallel TRC GUI window and login. While both TRC GUI windows can display the test request, only one user can control it (ensuring no duplicate control). In this way, two or more users can exchange control of the test request. By design, this does not require approval from the earlier instance of the TRC program. If you lose control of the test request, you can infer that someone else has logged into TRC and has control.

Note:

Virtual Network Computing (VNC) is a type of remote control software that allows control of another computer or device over a network.

3.4 Exiting the TRC

To exit the **TRC** window, click on the X button at the top-right of the window. Refer to *Figure 1* on page 4.

Important:

Accidentally closing the **TRC** window does not impact the state or progress of a test request in progress. This is because the TRC GUI does not execute the test request directly. The GUI allows viewing of the test request being executed by other programs within CyFlex. After the interruption, restart the TRC GUI and log in. The test request remains as before the interruption except for normal progress in an executing test.

4 Navigating the TRC Window

This describes the **TRC** window layout and functions.

Once a test request is loaded, several control buttons become active (no longer “grayed out”) to enable subsequent steps. Refer to *Section 5 Loading a Test Request* on page 12.

Figure 3: Annotated TRC Window

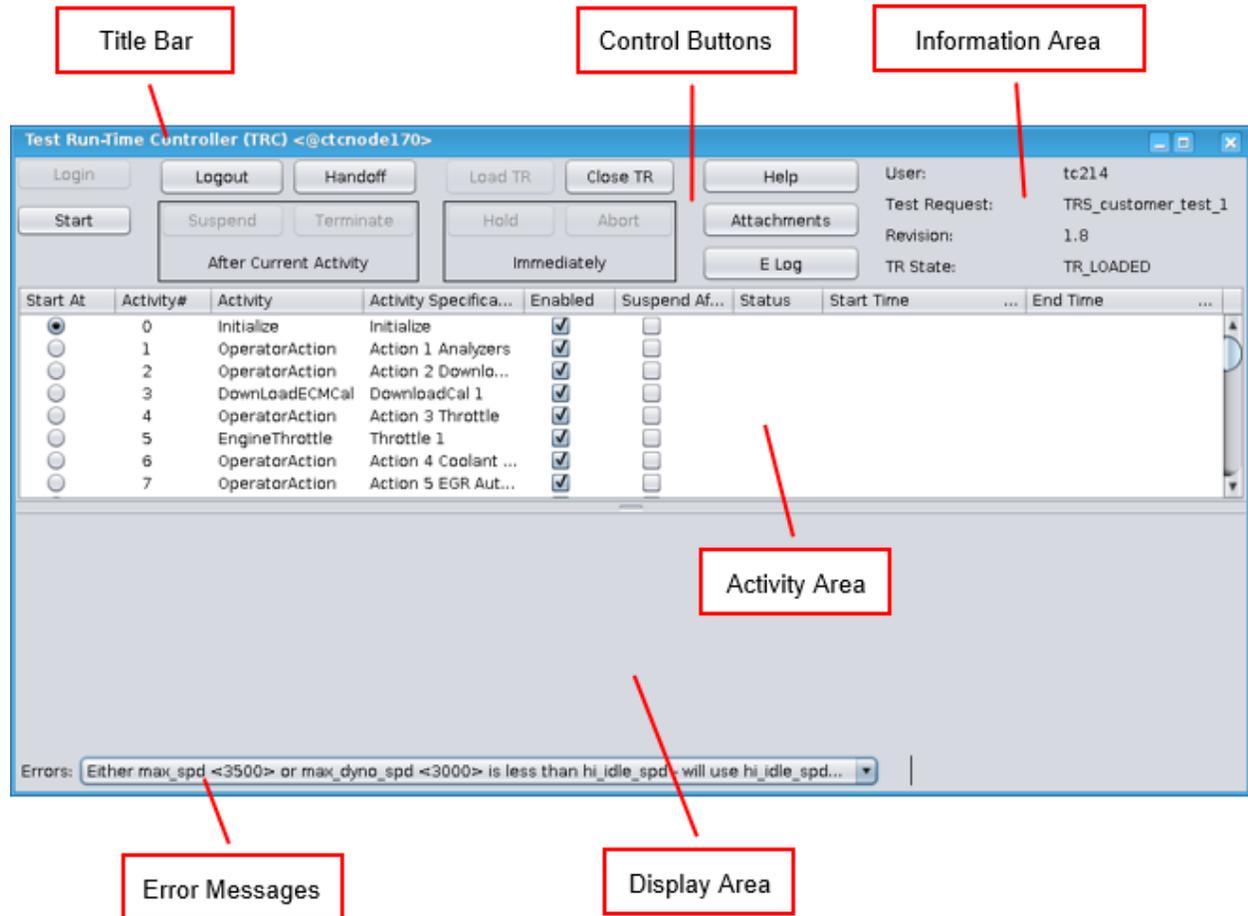
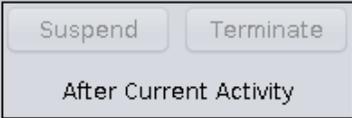


Table 1 describes **TRC** window components and functions.

Table 1: TRC Window Components and Functions

Control or Area	Description
Title bar	The title bar shows the TRC program and the name of the computer (node) on which it is running. Refer to <i>Figure 3</i> .
Login	 <p>Several control buttons appear below the title bar, including the TRC Login located in the top-left of the window.</p>

Control or Area	Description
<p>Logout</p>	 <p>Click this button to log out of TRC.</p>
<p>Handoff</p>	 <p>Click to logout and allow another user to login in one step.</p>
<p>Load TR</p>	 <p>Click to load a test request (TR) into TRC. Refer to <i>Section 5 Loading a Test Request</i> on page 12.</p>
<p>Close TR</p>	 <p>Click to close a test request that was previously loaded in TRC but is no longer needed at the test bed. Refer to <i>Section 7 Closing a Test Request</i> on page 18.</p>
<p>Start/ Resume/ Release</p>	 <p>The button label changes among Start, Resume, and Release depending on the available actions for the test request:</p> <ul style="list-style-type: none"> • Start appears if a test request is loaded but not executing. This control can start the test request at a specified activity or after an interruption. • Resume appears if the test request is loaded but suspended. This continues the test request from where it was suspended. • Release appears if the test request is loaded but held. This control releases the test request from where it was held. <p>Note: Refer to <i>Section 6 Executing a Test Request</i> on page 15 for description of “Suspended” and “held” test requests.</p>
<p>Suspend After Current Activity</p>	 <p>While a test request is executing, clicking the Suspend button allows the current activity to continue until it is finished. At that time, the test request is put into a suspended state and awaits further action from the operator. This feature allows the operator to perform an action between the activities specified in the test request and to continue the test request without any changes to their sequence.</p>

Control or Area	Description
<p>Terminate After Current Activity</p>	<p>This is similar to the Suspend After Current Activity control. The difference is that at the end of the current activity, execution of the test request is terminated. Note that after the test request is terminated, the operator can close the test request or restart the test request at any location (activity) in the test request at the beginning where it was terminated or at some other point. For that reason, the Terminate control is more flexible than the Suspend control.</p>
<p>Hold Immediately</p>	<div data-bbox="440 575 794 695" data-label="Image"> </div> <p>This is similar to the Suspend After Current Activity control, but the current activity is “frozen” immediately (to the extent possible). Refer to <i>Section 6 Executing a Test Request</i> on page 15 for additional information.</p>
<p>Abort Immediately</p>	<p>Click this button to cause the current activity to terminate immediately, to the extent possible. Refer to <i>Section 6 Executing a Test Request</i> on page 15 for additional information.</p>
<p>Help</p>	<div data-bbox="436 978 639 1031" data-label="Image"> </div> <p>Click this button to open a browser window with Help for using TRC.</p>
<p>Attachments</p>	<div data-bbox="436 1113 639 1165" data-label="Image"> </div> <p>All attachments (files referenced within the test request) are stored in a common directory. Clicking the Attachments button opens a file browser pointing to this directory. The user can examine the contents of these files. He or she can view at any time, for example, a schematic diagram referenced in an operator action or any file attached by the test request author.</p> <p>It is acceptable to add files to this directory. When the test request is closed, all the files in the Attachments directory, as they existed at the time, are saved back to the file-share where test requests are kept. This means the user should save and close any files that might have been edited before closing the test request.</p> <p>Note: The directory containing the attachments is also mapped to the Calterm Windows computer that exists in many engine test beds at Y:\TestRequest\.</p>
<p>E Log</p>	<div data-bbox="436 1709 639 1761" data-label="Image"> </div> <p>Clicking this control to open the E Log viewer and editor. This is for convenience. TRC does not make any entries in the E log.</p>

Control or Area	Description																																																																
<p>Information area – user and test request</p>	<div data-bbox="440 306 993 499" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <p>User: tc214</p> <p>Test Request: TRS_customer_test_1</p> <p>Revision: 1.8</p> <p>TR State: TR_LOADED</p> </div> <p>An Information area appears in the top-right of the TRC window, and displays:</p> <ul style="list-style-type: none"> • User: Available if an account is logged in to the TRC GUI • Test Request: Listed if a test request is loaded • Revision: Shown if a test request is loaded • TR State: The state of TRC and the test request: <ul style="list-style-type: none"> ○ NO_TR_LOADED ○ TR_LOADED ○ TR_EXECUTING ○ WAITING_TO_SUSPEND ○ TR_SUSPENDED ○ WAITING_TO_TERMINATE ○ TR_HELD 																																																																
<p>Activity area</p>	<div data-bbox="440 1073 1416 1308" style="border: 1px solid black; padding: 5px; margin-bottom: 10px;"> <table border="1"> <thead> <tr> <th>Start At</th> <th>Activity#</th> <th>Activity</th> <th>Activity Specific...</th> <th>Enabled</th> <th>Suspend ...</th> <th>Status</th> <th>Start Tim</th> </tr> </thead> <tbody> <tr> <td><input checked="" type="radio"/></td> <td>0</td> <td>Initialize</td> <td>Initialize</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>1</td> <td>Operat...</td> <td>Action 1 Analyz...</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>2</td> <td>Operat...</td> <td>Action 2 Downlo...</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>3</td> <td>DownL...</td> <td>DownloadCal 1</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>4</td> <td>Operat...</td> <td>Action 3 Throttle</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>5</td> <td>Engine...</td> <td>Throttle 1</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> <tr> <td><input type="radio"/></td> <td>6</td> <td>Operat...</td> <td>Action 4 Coolan...</td> <td><input checked="" type="checkbox"/></td> <td><input type="checkbox"/></td> <td></td> <td></td> </tr> </tbody> </table> </div> <p>The Activity area (pane) in the center of the TRC window shows activities for the test request in table format. TRC optimizes the layout of the table within the window space available.</p> <p>To adjust the pane size, use the mouse cursor to grab a side or corner of the pane and drag it. If the list of activities does not fit in the available space, a scroll bar appears at the right edge of the pane. Similarly, the border between the Activity area (pane) and the Display area below it can be adjusted. The horizontal borders between the columns are also adjustable.</p> <p>The following describes each table column in the Activity area:</p> <ul style="list-style-type: none"> • Start At: This is a column of radio buttons. Only one button can be selected at a time. The associated row for the radio button is the activity that the test request starts with (executes next). The activity chosen must be confirmed later and may be changed then if needed. • Activity#: This is a sequential number associated with each activity. The activities starting with 1 were specified by the test request author. 	Start At	Activity#	Activity	Activity Specific...	Enabled	Suspend ...	Status	Start Tim	<input checked="" type="radio"/>	0	Initialize	Initialize	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	1	Operat...	Action 1 Analyz...	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	2	Operat...	Action 2 Downlo...	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	3	DownL...	DownloadCal 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	4	Operat...	Action 3 Throttle	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	5	Engine...	Throttle 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>			<input type="radio"/>	6	Operat...	Action 4 Coolan...	<input checked="" type="checkbox"/>	<input type="checkbox"/>		
Start At	Activity#	Activity	Activity Specific...	Enabled	Suspend ...	Status	Start Tim																																																										
<input checked="" type="radio"/>	0	Initialize	Initialize	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	1	Operat...	Action 1 Analyz...	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	2	Operat...	Action 2 Downlo...	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	3	DownL...	DownloadCal 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	4	Operat...	Action 3 Throttle	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	5	Engine...	Throttle 1	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												
<input type="radio"/>	6	Operat...	Action 4 Coolan...	<input checked="" type="checkbox"/>	<input type="checkbox"/>																																																												

Control or Area	Description
	<p>Activity 0 (Initialize) is a special activity automatically added by TRS to get the test bed and/or unit under test into a known state for subsequent activities. By default, Initialize is the starting activity.</p> <ul style="list-style-type: none"> • Activity: Each activity in the test request has a name. Click a listed activity to view more information. Associated Help is displayed in a browser. • Activity Specifications: This is the name of the specifications associated with the activity. These are defined in the Test Request Interface (TRI) and customize each activity. To view the test specifications for an activity, right-click on the specifications name in the table column next to the associated activity name. TRI opens in a “read only” mode, with the selected activity specification as the initial focus. Values and options appear but cannot be changed in this mode. If fixing values is necessary, open TRI in normal mode on a CyFlex computer, make the changes, and resubmit the test request. Refer to the <i>Test Request Interface User Guide</i>. <p>Note Because re-submitting a test request changes the revision number, it is then necessary to close the current test request / revision in TRC and reload the new test request revision.</p> <div style="border: 1px solid black; padding: 5px; margin: 10px 0;"> <p>Important: The field values in each of the table columns described below save to the file-share when the test request is closed. This includes Enabled, Suspend After, Status, Start Time, and End Time. Loading the same test request and revision later, in the same or another test bed, begins with the saved field values. For example, if the operator closes a test request with a particular activity enabled, that same activity is enabled upon loading the request.</p> </div> <ul style="list-style-type: none"> • Enabled: This is a column of check boxes. Any number of activities may be checked (enabled). Each enabled is executed in the normal order listed. When TRC encounters an activity that is not enabled, it skips the activity and goes to the next. By default, new test requests have all activities enabled. The operator may disable or re-enable any of them using the check boxes, any time before executing the activity. Disabling an activity while it is executing does stop it. • Suspend After: This is another column of check boxes. Any number of activities may be checked (enabled). By default, new test requests have none checked. The operator may check or uncheck any of the boxes before the corresponding activity completes. When an activity completes and has Suspend After checked, the test request is placed in a suspended state, as if the operator used the Suspend After Current Activity control button. The advantage of using this column instead of the control button is that any number of suspended activities may be added before starting the request or during its execution. • Status: This column indicates the state of activities. The table row for an activity changes colors based on the activity’s state. Possible

Control or Area	Description															
	<p>states are:</p> <table border="1" data-bbox="544 331 1315 630"> <thead> <tr> <th><u>Field value</u></th> <th><u>Line color</u></th> <th><u>Meaning</u></th> </tr> </thead> <tbody> <tr> <td>Blank</td> <td>White</td> <td>Execution not started</td> </tr> <tr> <td>In-progress activity</td> <td>Cyan</td> <td>Currently executing the</td> </tr> <tr> <td>Success successfully</td> <td>Green</td> <td>Previously executed</td> </tr> <tr> <td>Failure unsuccessfully</td> <td>Pink</td> <td>Previously executed</td> </tr> </tbody> </table> <p>If an activity is re-executed, the new value and color overwrite the previous ones.</p> <ul style="list-style-type: none"> • Start Time: Indicates the date and time when each activity was started. If an activity is re-executed, the new date and time overwrite the old values. • End Time: Shows the date and time when each activity ended If an activity is re-executed, the new date and time overwrite the old. 	<u>Field value</u>	<u>Line color</u>	<u>Meaning</u>	Blank	White	Execution not started	In-progress activity	Cyan	Currently executing the	Success successfully	Green	Previously executed	Failure unsuccessfully	Pink	Previously executed
<u>Field value</u>	<u>Line color</u>	<u>Meaning</u>														
Blank	White	Execution not started														
In-progress activity	Cyan	Currently executing the														
Success successfully	Green	Previously executed														
Failure unsuccessfully	Pink	Previously executed														
Display area	<p>The Display area is located in the TRC window below the Activity area. When a test request is executing, this area shows a limited set of CyFlex variables related to the activity.</p> <p>Note: The CyFlex variables displayed are determined by TRS support personnel, and because those chosen will apply to all test beds, cannot be customized for each test bed.</p>															
Error Messages	<div data-bbox="440 1220 1408 1251" style="border: 1px solid gray; padding: 2px;"> Errors: Warning: no updates on list ECM0 killing stream handler.Run asam3_specs to restart. </div> <p>This is located at the bottom of the TRC window. Normally, it displays the last error message generated within CyFlex.</p> <p>To view previous errors, click on the arrow at far right in the message area. The last five errors appear.</p>															

5 Loading a Test Request

A test request can be loaded only if another is not already loaded. If a test request is currently loaded, the **Load TR** button is inactive.

Execute the following steps to load a test request:

1. Click the **Load TR** button to display the **Test Request Select/Search** window as in *Figure 4*.

Figure 4: Test Request Select/Search Window

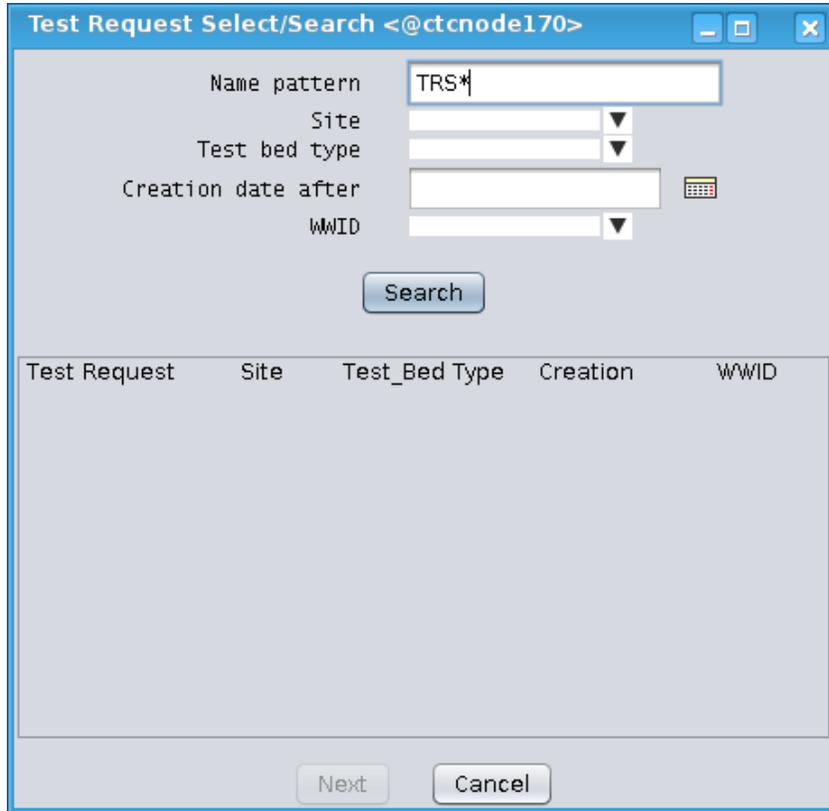


Table 2 describes the fields and functions in the **Test Request Select/Search** window.

Table 2: Test Request Select/Search Fields

Field	Description
Name pattern	The TRC retains the most recently loaded test request in the Name pattern field. Either accept the retained Name Pattern or specify a different test request. As needed, enter known parts of the name combined with a wild card character such as an asterisk (*) or question mark (?). For example, use "*" to match zero or more characters. Use "?" to match just one character.
Site	Click the drop-down and select the Site for which the test request was created. Example: CTC

Field	Description
Test bed type	Click the drop-down and select the Test bed type for which test request was designed. Example: CTC EML
Creation date after	Click the calendar icon to specify a date. This defines the earliest date when the test request was created
WWID	Click the drop-down and select the WWID representing the identity of the person who created the test request.

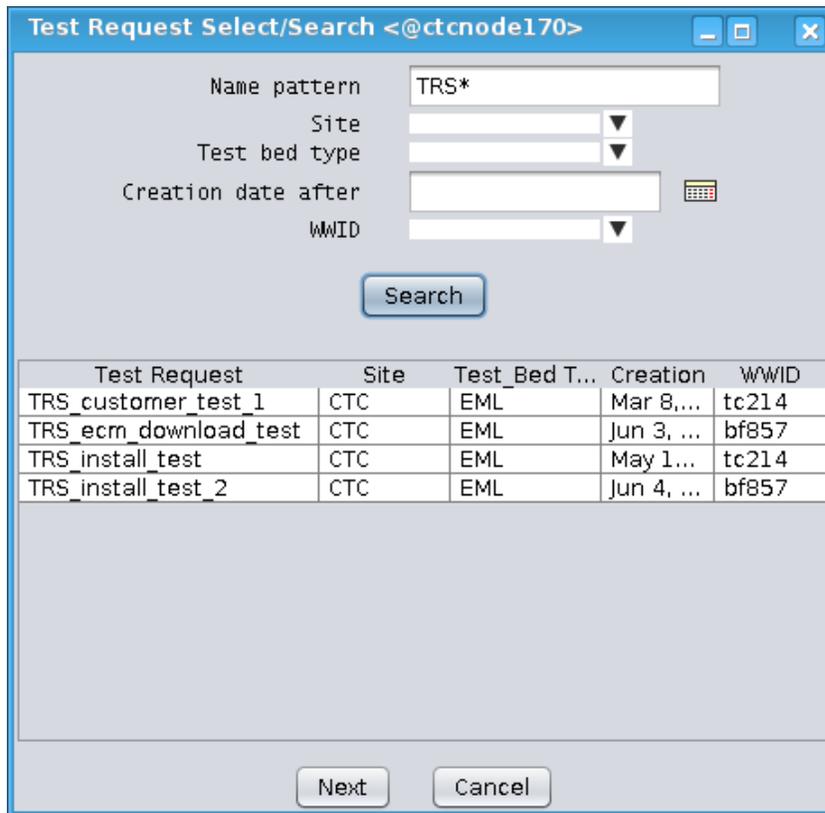
Note:

Fields left blank will match any value for the field. Entering values in multiple fields will match only those test requests that meet all the criteria.

2. Click the **Search** button

The search may take a moment. TRC searches the database of test requests and returns test request names that match the criteria as shown in *Figure 5*.

Figure 5: Returned Search Results



3. Select a **Test Request** from the list and then click the **Next** button. TRC searches the database and returns a list of revisions for the test request as in *Figure 6*.

Figure 6: Returned Test Request Revision List

Revision	Site	Test Bed Type	Creation	WWID
1.8	CTC	EML	Jun 1, 2015	tc214
1.7	CTC	EML	May 26, 2015	tc214
1.6	CTC	EML	May 13, 2015	tc214
1.5	CTC	EML	Apr 3, 2015	tc214
1.4	CTC	EML	Mar 12, 2015	tc214
1.3	CTC	EML	Mar 11, 2015	tc214
1.2	CTC	EML	Mar 10, 2015	tc214
1.1	CTC	EML	Mar 9, 2015	tc214
1.0	CTC	EML	Mar 8, 2015	tc214

@Notes:

When revisions populate the list, the **Next** button activates. If only a single item is in the list, that item is automatically “selected.”

As needed, select the **Previous** and **Cancel** buttons throughout the steps to:

- Return to the previous step
- Cancel the Load TR request

4. Load the test request revision. Select a **Revision**.
5. Click the **Next** button. The following factors may affect time to process:
 - Size and complexity of the test request
 - If the test request/revision was used before in a test bed, loading is usually fast.
 - If the test request/revision was not used before, loading takes longer. This is because TRC must execute a support program to translate the test request into a set of files for the CyFlex system.
6. When the loading process completes, the Activity area of the **TRC** window is populated and the **Start** button active to indicate the test request is ready for use.

6 Executing a Test Request

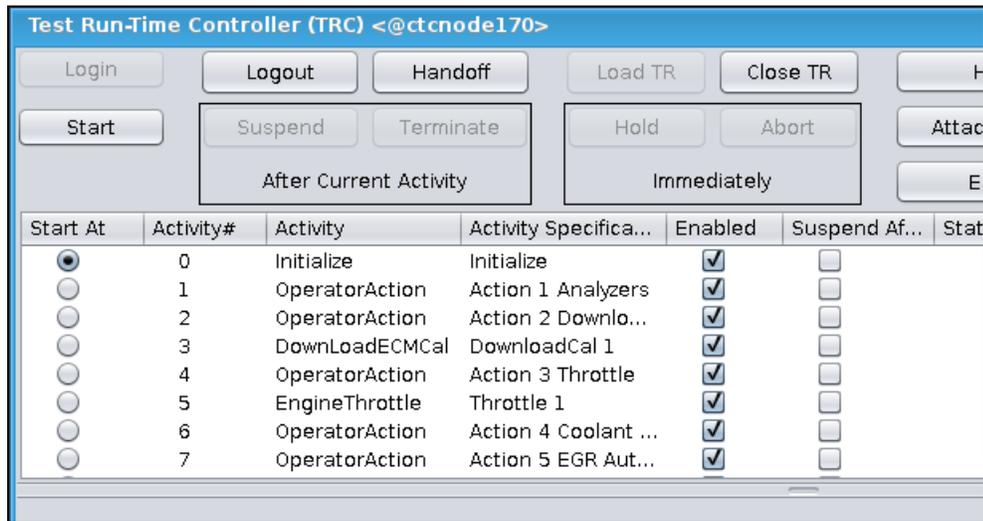
Once logged in at the TRC window, the operator has full control for executing the test request.

6.1 Starting or Restarting a Test Request

Execute the following steps to start a test request:

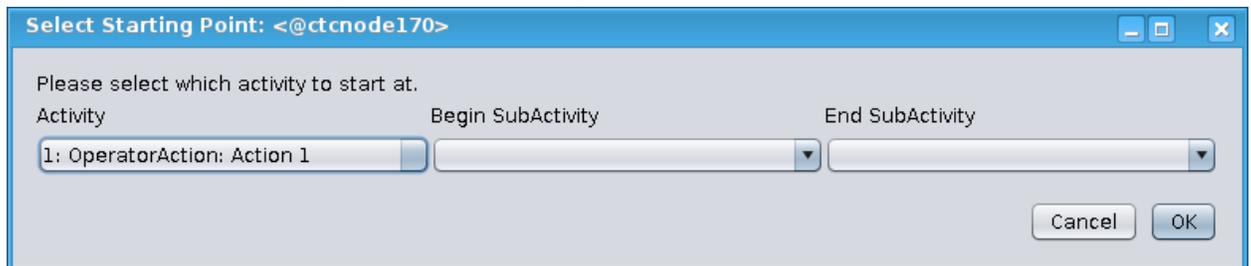
1. In the Activity area of the TRC window, select the radio button in the **Start At** column on the row corresponding to the first activity to be executed as in *Figure 7*.

Figure 7: Select a Test Request Activity



2. Check each activity to perform in the **Enabled** column.
3. Check applicable boxes in the **Suspend After** column
4. Click the **Start** button to display the **Select Starting Point** window as in *Figure 8*.

Figure 8: Select Starting Point Window



Perform the following as warranted:

- Click **Cancel** to cancel starting the test request if the **Start** button in the TRC window was accidentally clicked
- Change the starting activity if it was incorrectly set by selecting a different activity from the **Activity** drop-down menu
- For selected tests, change the **Begin SubActivity** and **End SubActivity**
- Click **OK** to incorporate the selected starting point.

The **SubActivity** control buttons are supported only for certain activities. An example of supported activities might include steady state mapping driven by a test table having many operating conditions. Sub-activities are represented by lines in the test table for the various conditions.

If the test request was previously executed and aborted, but a number of sub-activities were successfully executed, the test can be restarted on the activity not yet completed and those already done skipped. Also, an activity can be stopped before the end of the test table.

By default, the beginning sub-activity is the first, and the ending sub-activity the last. To set these, click on each field per the example in *Figure 8* on page 15 and select the sub-activity. This feature is currently available only for the activity that starts the test request.

However, if a subsequent activity further down in the activity table requires this feature, an option is to set the test request to suspend the prior activity. This means, use the **Suspend After** check box to suspend the activity that occurs before the one to use the sub-activity feature. When the test request suspends, click the **Resume** button to display the choice of sub-activities again. Refer to the description of the **Suspend After** on page 10.

6.2 Suspending after the Current Activity

Execute the following steps to perform some other task immediately after the current activity in progress completes:

1. Click the **Suspend After Current Activity** button in the **TRC** window. The test request can then be resumed later.

The **TR State** in the top-right of the **TRC** window changes to **WAITING_TO_SUSPEND**. When the current activity is complete, the test request is suspended and the **TR State** changes to **TR_SUSPENDED**.

2. Perform the other task. This may be a non-emergency inspection that is considered appropriate by the operator on the scene, but not anticipated by the engineer who created the test request. For example, checking the test bed for any leaks that are not an emergency.

The label on the **Start / Resume / Release** button changes to **Resume**.

Warning: Be aware of any test in progress that could make the test bed area unsafe. Wait until the current test activity is completed and the test suspended before entering the test bed area.

3. Click the **Resume** button to continue with the test request that was suspended. The next enabled activity starts.

6.3 Terminating after the Current Activity

Execute the following steps to stop the test request immediately after the current activity in progress:

1. Click the **Terminate After Current Activity** button.

The **TR State** changes to **WAITING_TO_TERMINATE**.

When the current activity is completed, the test request stops and the **TR State** changes to **TR_LOADED**.

2. Continue the test request at the next activity or a different activity in the same request. Refer to *Section 6.1 Starting or Restarting a Test Request* on page 15.

6.4 Holding the Current Activity

Execute the following steps to hold (pause) the current activity (to the extent possible):

1. Click the **Hold Immediately** button. The test request can be continued later.

The **TR State** changes to **TR_HELD**.

The label on the **Start / Resume / Release** button changes to **Release**.

Note:

The script executing the activity is “frozen” in place and the activity does not proceed. However, some aspects of the activity in progress might be beyond the control of TRC. For example, a “calibration download” to ECM by CUTY, once initiated by CyFlex, is beyond control of the TRC until the download is finished. For questions, consult with CyFlex support personnel at SGS

2. Click the **Release** button to release the activity from hold.

6.5 Aborting the Current Activity

Execute the following steps to abort functions of the current activity and stop the test request:

1. Click the **Abort Immediately** button.

The script executing the activity ends regardless. The Test Manager (`gp_test`) program instance is terminated by execution of an idle command.

The **TR State** changes to **TR_LOADED**.

Warning / Caution: While this is safe in most conditions, the state of the unit under test after aborting the activity depends on the activity. If unsure, consult SGS before using the **Abort Immediately** control button.

2. Continue the test request. Refer to *Section 6.1 Starting or Restarting a Test Request* on page 15.

7 Closing a Test Request

After executing a test request, and if no more work is needed using the test request, it may be closed.

Closing a test request causes TRC to take a “snapshot” of the test request, including the states of all of its activities, and returns it along with all of its attachments to the file-share from where it was originally obtained. If the test request was not completed for some reason, this feature allows for the test request to be continued in the same or an equivalent test bed later.

Before closing a test request:

- Confirm the **TR State** is **TR_LOADED**.
The **Close TR** button remains inactive until the **TR State** is correct.
- Save (if appropriate) and close any file being edited from the attachments directory.

Click the **Close TR** button to close a test request.

@Note:

There is a brief delay before the files are queued to be sent back to the file-share. In order to work even if the computer network between the test cell computer and the file-share is temporarily down, the queuing system will retry until the network is back in operation and the files can be saved.