Tester Toolkit User Guide

04 MAR 2014 - Version 38

Radical Blue Gaming, Inc.
85 Keystone Avenue Suite F
Reno, Nevada 89503

call us: +1.775.329.0990
visit us: www.radblue.com
drop us an email: sales@radblue.com

Need help?

At the RadBlue forum you can find the latest release information, report issues, get your questions answered, and submit suggestions for improving our products. Simply log on to: http://radblue.mywowbb.com

Find out more about the GSA protocols

If you want to find out more about the Gaming Standards Association and the work being done in the area of protocol standardization for the gaming industry, we encourage you to visit their website at www.gamingstandards.com.

radblue

Contents

About Startup Algorithms

Startup Algorithms let you define the commands sent to the EGM when communications are initiated. You can choose to run the RadBlue read-only startup algorithms to completion, or you can use them as a template to create a custom startup configuration for your own testing.

If you modify the startup you can select the step-through option to manipulate the startup timing, one command at a time. Once you create a startup algorithm, you can modify the commands and their sequence, as needed, or delete the algorithm.

All default and newly created algorithms are saved to the following location: [**RGS installation directory**] > **custom-config** > **startup**

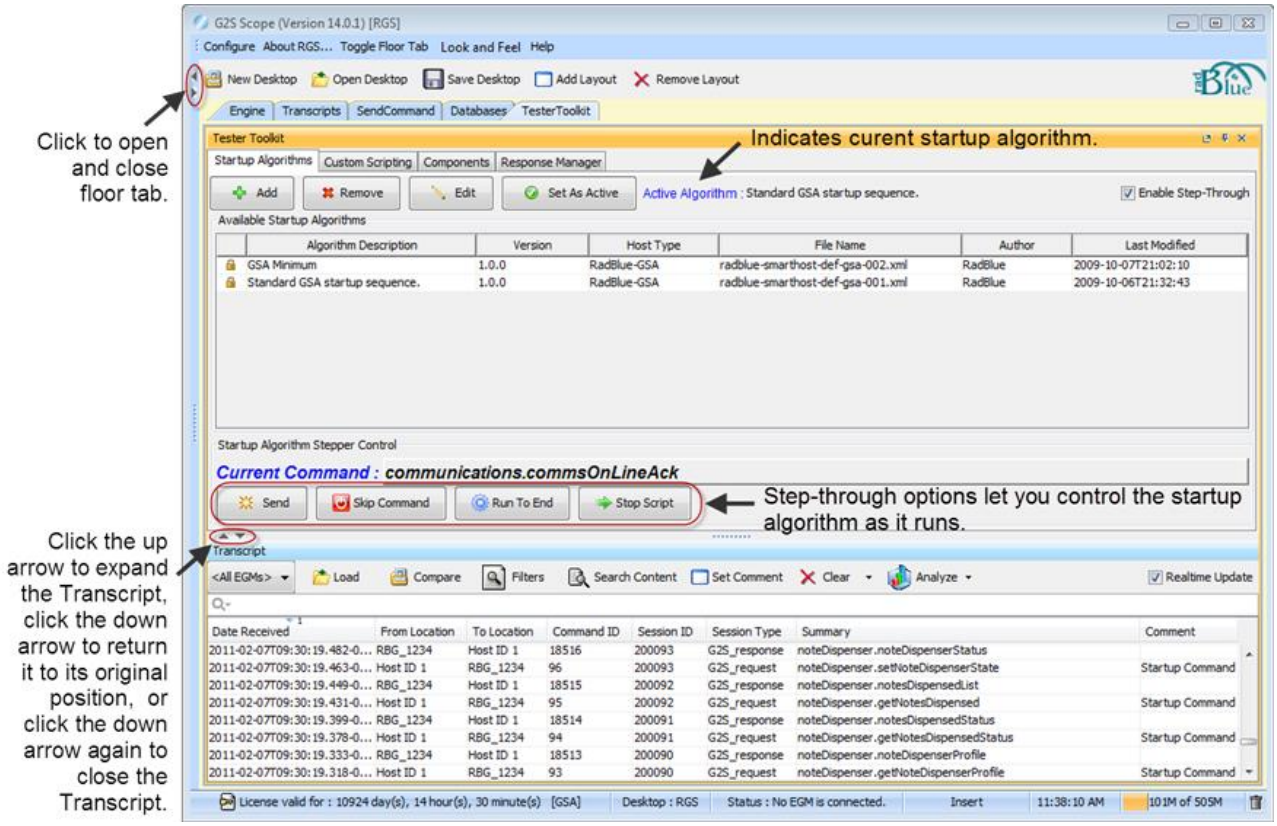
Unless you are familiar with G2S startup communications as defined in the Gaming Standards Association's G2S Message Protocol document we recommend that you use a default startup algorithm (**select default** > **Edit** > **modify** > **Save As**) when creating new algorithms. In addition, be sure to read the following sections for detailed information on creating startup algorithms:

- Set Up a Custom Algorithm
- Work with the Custom Algorithm Controls
- Add a Command Group to a Custom Algorithm

The Transcript appears at the bottom of the Startup Algorithms layout so you can see how messages flow between RGS and the EGM as the algorithm executes.

Review the Startup Algorithms Layout

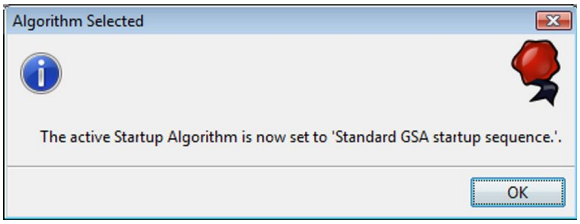
The Startup Algorithms tab is located on the Tester Toolkit layout. The Startup Algorithms tab contains the Startup Algorithm object and the Transcript object.



Set Active Startup Algorithm

Before you can use a custom startup algorithm, you must set the active algorithm.

1. From the **Tester Toolkit** layout, select the **Startup Algorithms** tab.
2. Select the algorithm you want to set as active from the **Available Startup Algorithms** list.
3. Click **Set As Active**.



4. Read the informational message, and click **OK**.
5. Verify that the **Active Algorithm** field displays the description of the algorithm you selected in step 1.

Add or Edit an Algorithm

Follow these steps to create a new custom algorithm or edit an existing algorithm.

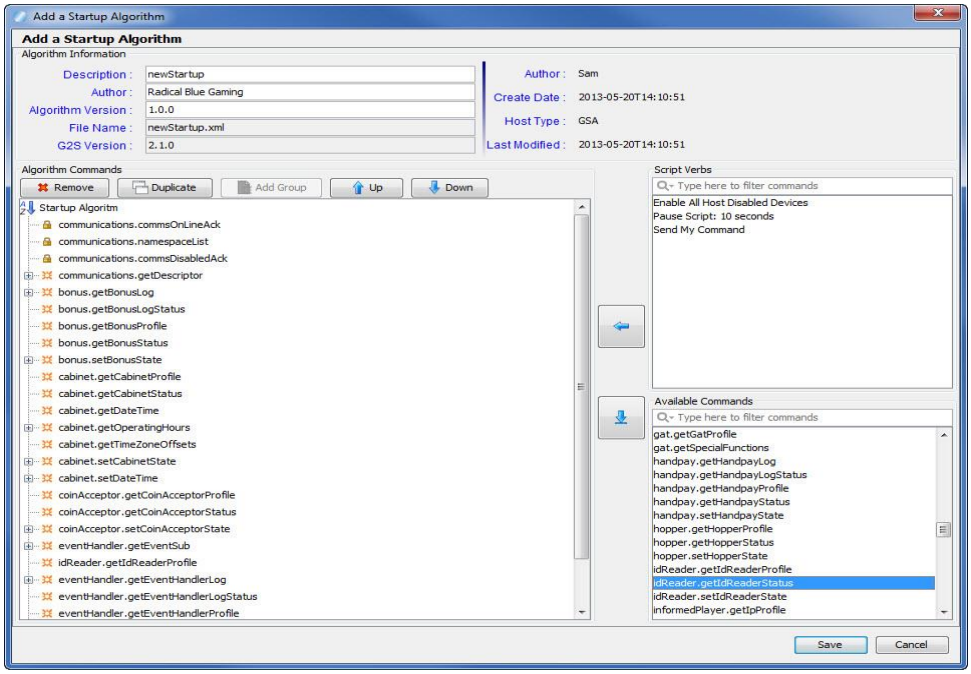
1. Go to: **Tester Toolkit** > **Startup Algorithms**
2. To create a new custom algorithm:
   a. Click **Add**.

   b. Type a file name for the algorithm (spaces are not allowed). Ensure the algorithm file name is unique.
   c. Click **Apply**.

or

To open an existing algorithm to edit:

   a. Highlight the algorithm you want to edit.
   b. Click **Edit**.



3. Modify information for the algorithm.

   Note: RGS automatically inserts default information for new algorithms.

   - **Description** - Type a description for the algorithm.
   - **Author** - Type an identifier for the algorithm's author.
   - **Algorithm Version** - Type a version number for the algorithm. This field can contain any information required.
   - **File Name** - File name of the algorithm (entered in step 2b). This field is *read-only*.
   - **G2S Version** - Indicates the selected version of the Game To System (G2S) protocol. The G2S version is defined in the Schema Option field under **Configure** > **Engine Options**. This field is *read-only*.

4. Clear the **Verbose Algorithm** checkbox if you do not want RGS to send the `get[device]Status` and `set[device]State` commands for devices that appear as *hostEnabled* in the `descriptorList` command. By default, this option is selected.
5. Add, delete and modify commands and command groups as required.

   See Set Up a Custom Algorithm, Work with the Custom Algorithm Controls, and Add a Command Group to a Custom Algorithm for more information.

   You can add script verbs as required. For more information, see Enable All Host Disabled Devices, Pause Script and Send My Command.

6. Click **Save** or **Save As**, as required, to save the algorithm. All saved algorithms display in the *Available Startup Algorithms* list.

   If you choose **Save As**, remember that the algorithm name must be unique.

Set Up a Custom Algorithm

According to the G2S specification, to begin communications with an EGM, the host must send the `communications.commsOnLineAck` and `communications.commsDisabledAck` commands in response to the EGM sending the `communications.commsOnLine` and `communications.Disabled` commands.

Once these commands have been sent, RGS can send additional commands to determine which devices the EGM supports, set the status of devices, and get device information. Finally, to enable communications with the EGM, the `communications.setCommsState` command should be sent with the enable attribute set to **true**.

Since you must send the `communications.commsOnLineAck` and `communications.commsDisabledAck` commands to begin the communications process, they appear at the beginning of every algorithm and cannot be removed. The

`communications.setCommsState` command, however, is optional and can be sent as your testing needs require.

Another optional, but important command that we recommend you put into your custom startup algorithm is the `communications.setKeepAlive` command. This command is used to set the frequency with which the EGM sends the `communications.keepAlive` command to RGS. The default interval is 30 seconds (30000 milliseconds). However, you can change the interval by going to **Configure** > **Engine Options** > **Keep Alive Interval**. Type a new interval in milliseconds and click **Apply**.

Work with the Custom Algorithm Controls

To create a custom startup algorithm, you move G2S commands from the Available Commands section on the right side of the screen to the Algorithm Commands section on the left side of the screen.

To move a command into an algorithm, click an Available Command to highlight it and click the left arrow. You can use CTRL+click and SHIFT+click functions to select and move multiple commands at once. Alternately, you can click the  button to add commands to the end of the Algorithm Commands list.

To quickly find a command, use the Quick Filter tool at the top of the Available Commands list. This feature allows you to filter messages based on entered data. Simply click inside the text box next to the magnifying glass and start typing. The command list dynamically filters as you type. For additional options, click the magnifying glass. A menu displays that you can use to provide additional selection criteria.

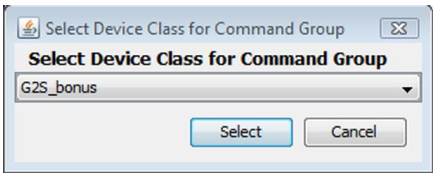
Once a command is in the Algorithm Commands section, you have several options for moving and modifying each command:

- Use the up and down arrows to move a selected command in the list.
- Click the plus ("+") sign to view command attributes. Note that only configurable command attributes are displayed and not all commands have configurable attributes.
- To configure an attribute, double-click the attribute, select or type the new value, and click **Save**.
- To delete a command from the list, click to highlight it and click **Remove**.
- To duplicate a command, including current settings for to associated attributes, click to highlight the command and click **Duplicate**.

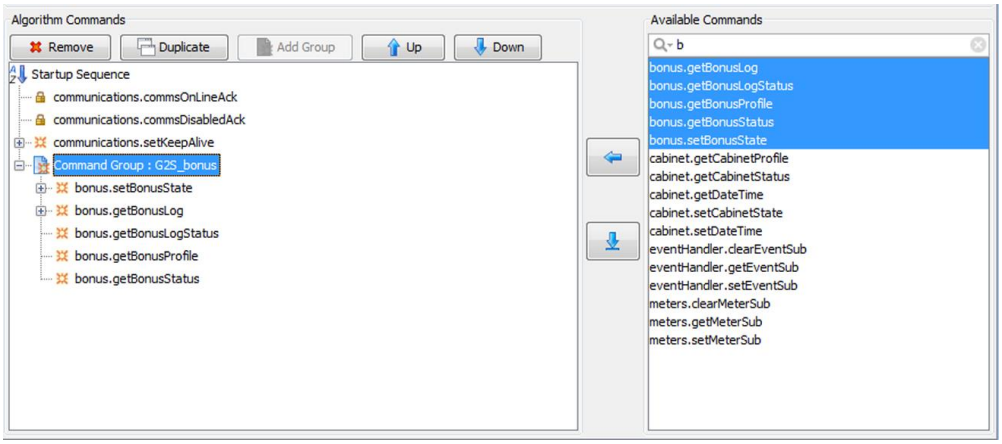
Add a Command Group to a Custom Algorithm

The Add Group option lets you send multiple commands in a single message. All commands in a group must be from the same class.

1. In the Algorithm Commands section, click to highlight **Startup Sequence** (at the top of the tree).
2. Click **Add Group**.



3. Click the drop-down arrow, and select a class for the new group. *Only commands from the selected class are allowed in the group* because the group gets sent to every device in the class.
4. To add commands to a group:
   a. Click to highlight the group.
   b. Select the command(s) you want to add from the Available Commands list.
   c. Click the left arrow to move the command(s) into the selected command group. For example:



5. Use the up and down arrows to move commands within the command group.
6. Save your changes. *RadBlue algorithms are read-only*, so you click must Save As to save your changes.

Step Through Your Startup Algorithm

The Startup Algorithm Stepper Control lets you send the commands in an algorithm one at a time, or not at all, to see how the EGM responds. The Startup Algorithm Stepper Control is located directly below the Available Startup Algorithms list on the Startup Algorithms tab.

For information on creating or editing a startup algorithm, see Add or Edit an Algorithm.

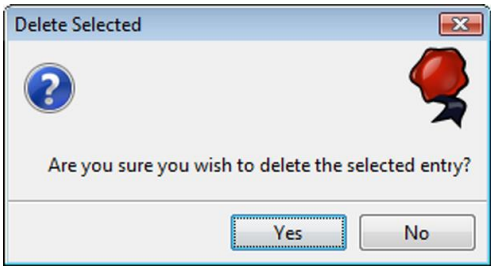
1. Go to: **Custom Configure** > **Startup Algorithms**
2. From the Available Startup Algorithms list, click to select the startup algorithm you want to use with the Startup Algorithm Stepper Control.
3. Click **Set As Active**.
4. Select **Enable Step-Through**.
5. *If RGS is currently communicating with the EGM*, go to the Send Command layout tab and click **Force MSX003** to force a restart of communications with the EGM. Otherwise, initiate communications from the EGM.
6. If you are not already there, return to the **Startup Algorithm Stepper Control** on the Startup Algorithms tab.

7. Look for the first command to appear in the Current Command field.
8. When a command appears in the Current Command field, take one of the following actions:
   - **Send** - Click to send the command to the EGM.
   - **Skip Command** - Click to not send the current command and to move to the next command in the algorithm.
   - **Run To End** - Click to send all commands, including the current command, until the end of the startup algorithm is reached.
   - **Stop Script** - Click to stop the startup algorithm. Any unsent commands will not be sent to the EGM.

Delete a Startup Algorithm

You can delete any algorithm that you create. RadBlue default algorithms (for example, GSA Minimum) are *read-only* and cannot be removed.

1. From the **Available Startup Algorithms** list, click to select the algorithm you want to delete.
2. Click **Remove**.

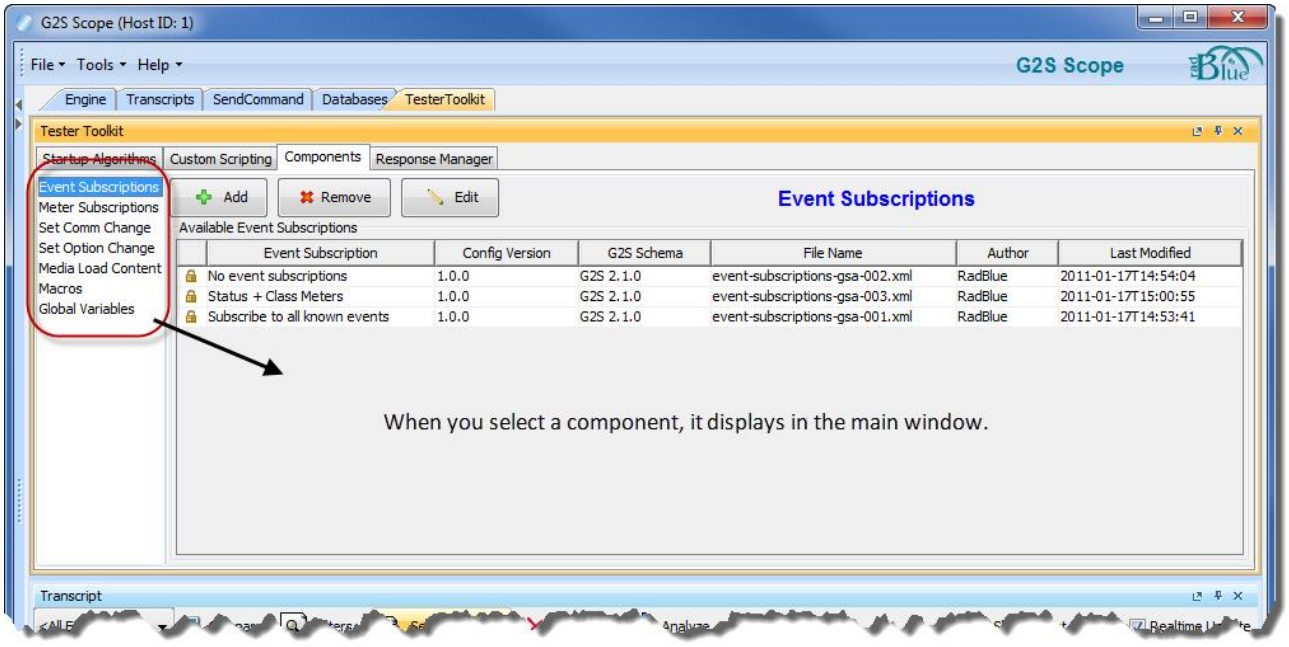


3. Click **Yes** to delete the selected algorithm, or click **No** to return to the main screen without making any changes.

About Components

The Components tab, located on the Tester Toolkit layout, lets you configure event subscriptions, meter subscriptions and macros. Macros are multi-command operations that can be run inside a custom script. You can also configure `commConfig.setCommChange` and `optionConfig.setOptionChange` commands for use in custom scripts and startup algorithms.

The Load Media Content component lets you define media content for the `mediaDisplay.loadContent` command that can also be used in custom scripts.

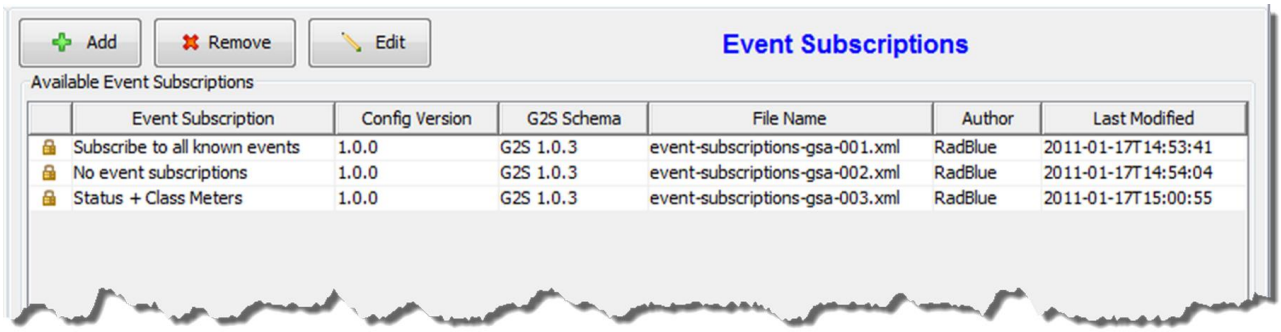


The following components can be added and/or edited directly from the Custom Scripting interface:

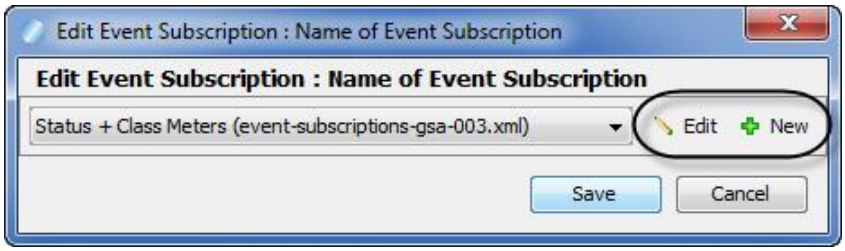
- Event Subscriptions
- Meter Subscriptions
- Set Comm Change
- Set Option Change
- Media Load Content
- Macros

Using Event Subscriptions

Event Subscriptions let you create event subscription files that can be used in the startup algorithm or custom scripts. Changes to event subscription files dynamically update anywhere they are used. For example, if you insert an event subscription file into a custom script (through the `eventHandler.setEventSub` command), and then make changes to it, the script is automatically updated.



You can add or edit a subscription from the Custom Scripting interface when you add the `eventHandler.setEventSub` command to a script.



Event subscriptions are additive so any new events will be added to the EGM's existing event subscription list. If you want to remove some or all event subscriptions, you must clear the event subscription list before you send a new event subscription.

Note that RGS only sends events that are supported by the EGM, including custom events. If the EGM does not support a particular event, it is discarded before the `setEventSub` command is sent.

Add Custom Events

By default, you can subscribe to all G2S events. However, you can also subscribe to custom and third-party events by editing the event set XML file.

> **Note:** RGS only sends events that are supported by the EGM, including custom events. If the EGM does not support a particular event, it is discarded before the `setEventSub` command is sent.

To add events, you must save the vendor event set file without the "-sample" extension into the schema directory used by the tool, and then make your changes. Note that RGS does not validate custom events.

1. Go to: [**installation directory**] > **schemas** > **g2s** > [**schema directory**]
2. Right-click the **g2s-event-set-vendor-sample.xml** file, and select **Edit**.
3. Go to: **File** > **Save As**, and navigate to the directory of the schema you'll being using with the tool.
4. Change the file name to **g2s-event-set-vendor.xml**, and click **Save**.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>

<events>

<event-group device-class="G2S_cabinet">

<event code="RBG_CBE001" text="The first Cabinet event." />

<event code="RBG_CBE002" text="The second Cabinet event." />

</event-group>

</events>
```

5. Modify the event name, and add new command name attributes, as needed.
   In the G2S validation manager file, there are two attributes:

   - **device-class**
     Use the device-class attribute to define the class you want associated with the custom event.
     The device-class attribute also serves as a grouping for events in the user interface. For example, in the following scenario, the you would find the RBG_BKE001 event under a "backgammon" node in the event tree on the Event Subscriptions dialog:

     ```
     <event-group device-class="G2S_backgammon">
     <event code="RBG_BKE001" text="The first backgammon event." />
     ```

   - **event code**
     Use the event code sub-element to specify each event you want to add.

6. Select **File**, and click **Save** to save your changes.
7. If the tool is running, stop it.
8. Start the tool.
9. Go to: **Tester Toolkit** > **Components** > **Event Subscriptions**
10. Add or edit an event subscription, and verify that the event(s) you added now appear in the event tree on the Event Subscriptions dialog.

Add or Edit an Event Subscription

Although pre-configured event subscriptions are available, you can also add custom event subscriptions. Once added, you can edit those subscriptions as needed.

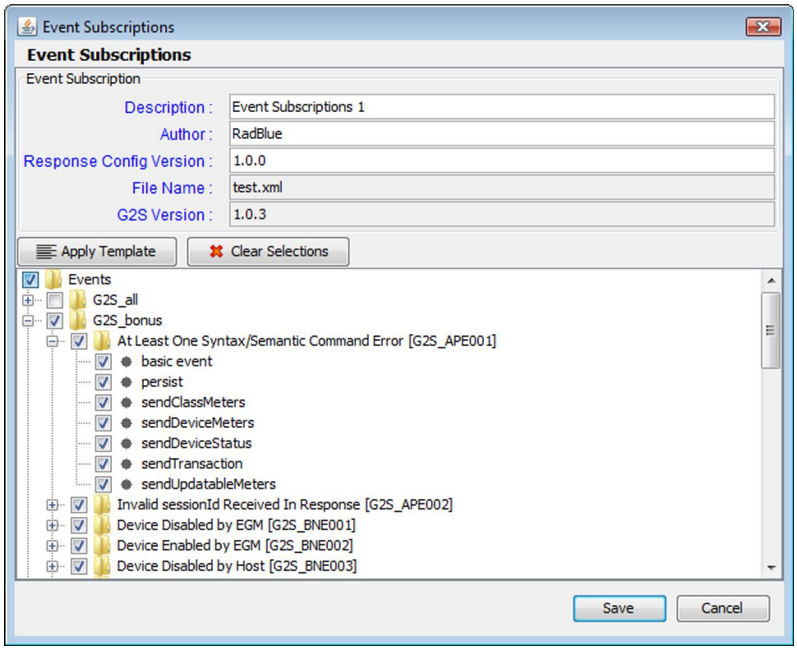
1. Go to: **Custom Configuration** > **Components** > **Event Subscriptions**
2. To create a new event subscription:
   a. Click **Add** to add a new event subscription.

   b. Type a name and description for the new event subscription.
   c. Click **Continue**.

or

To open an existing event subscription to edit:

   a. From Event Subscriptions, click to select the event subscription you want to edit.
   b. Click **Edit**.



3. Modify information for the event subscription configuration. Note that RGS automatically inserts default information.
   - **Description** - Type a description for the event subscription.
   - **Author** - Type an identifier for the event subscription author.
   - **Event Config Version** - Type a version for number for the event subscription. This field can contain any information required.
   - **File Name** - File name of the event subscription. This field is read-only.
   - **G2S Version** - Indicates the selected version of the Game-to-System (G2S) protocol. The G2S version is defined in the Schema Option field under Configure > Engine Options. This field is read-only.
4. Configure the event subscription.
   a. Select the event or class of events you want to subscribe to when the RGS starts up. At a minimum, you must select basic event for each event you want to receive.

   or

   a. Use **Apply Template** to apply specific information types to all events.
   b. Click **Apply**.

   Note: To clear all selected event subscriptions and attributes, click **Clear Selections**.
5. If you are adding a new event subscription, click **Save**.

   or

   If you are editing an existing event subscription:

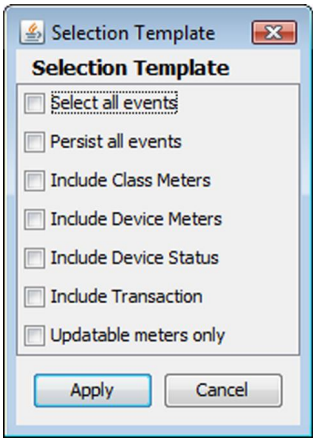
   a. Click **Save As**.

   b. Type a name and description for the new meter subscription.
   c. Click **Continue**.

Apply Template

Use the Apply Template option to apply specific types of event information to all events.

1. Click **Apply Template**.



2. Select the information you would like to receive from all events, or choose **Select all events**.
3. Click **Apply**.
4. Click **OK**.

Update Event Subscriptions

Whenever you modify event subscriptions, you must subscribe to the new events. If you do not subscribe to an event, the EGM will not report the occurence of that event to the RGS.

> **Note:** RGS only sends events that are supported by the EGM, including custom events. If the EGM does not support a particular event, it is discarded before the `setEventSub` command is sent.

Event subscriptions are additive so any new events will be added to the EGM's existing event subscription list. If you want to remove some or all event subscriptions, you must clear the event subscriptions before you update the event subscriptions.

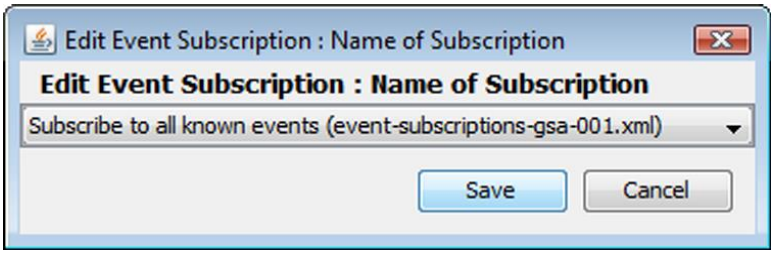
1. Select the **Custom Configuration** tab.
2. Select the **Startup Algorithm** tab.
3. Select the Standard GSA startup sequence startup algorithm. By default, this startup algorithm contains the `eventHandler.getSupportedEvent` and `eventHandler.setEventSub` commands.

4. Click **Set As Active**.
5. Now that you've configured the startup algorithm, click the Send Command tab.
   If you want to clear the current event subscription list before the new event subscriptions take effect:
   a. Select **G2S_eventHandler** > **clearEventSub**
   b. Click **Send Selected Command**.
   c. Click **Select All** to clear all events by setting the Selected value to **true** for all event subscriptions.
   d. To change individual events, click the **Selected** column of the event you want to change, click the drop-down arrow.
6. Restart communications with the EGM to run the startup algorithm.

Insert Event Subscription File into a Custom Script

You can add an event subscription file to a custom script simply by inserting the `setEventSub` command into the script. Note that RGS only sends events that are supported by the EGM, including custom events. If the EGM does not support a particular event, it is discarded before the `setEventSub` command is sent.

1. Go to: **Tester Toolkit** > **Custom Scripting**
2. Either click **Add** to create a new script, or select an existing script and click **Edit**.
3. Scroll to **eventHandler.setEventSub** in the Available Commands list.
4. Left-click and drag the command into the Custom Script Commands section.
5. Click the plus sign to expand the command information.
6. Double-click **Name of Subscription**.

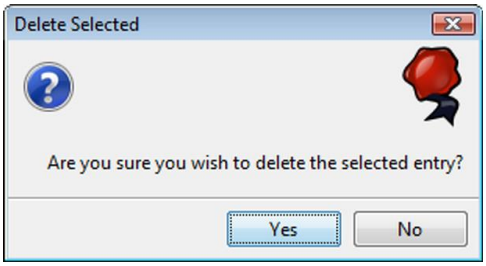


7. Click the drop-down arrow, and select the event subscription file you want to run at the point you have inserted the `setEventSub` command in the script.
8. Click **Save**.

Delete an Event Subscription

The Remove option lets you delete saved custom event subscriptions as needed. Note that pre-configured event subscriptions, denoted by a lock symbol to the left of the entry, cannot be deleted.

1. Click to select the event subscription you want to delete.
2. Click **Remove**.

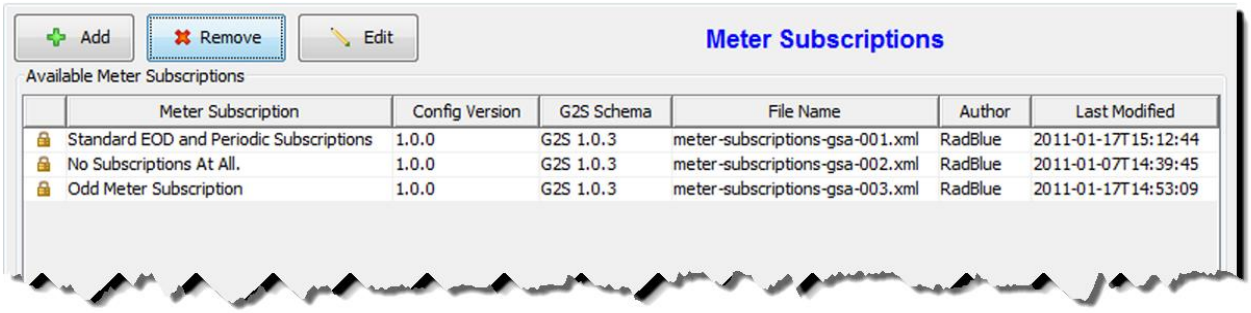


3. Click **Yes** to delete the selected event subscription.

Using Meter Subscriptions

Meter Subscriptions let you define which EGM meters (currency, device, game denomination and wager) you want to receive and the reporting interval (end-of-day and periodic). If you choose periodic reporting, you can configure the interval.

You can select whether to include meter definitions in meter reports.



You can add or edit a subscription from the Custom Scripting interface when you add the `meters.setMeterSub` command to a script.

For end-of-day (EOD) meters, you must specify the offset from midnight. This is the time that end-of-day meters are sent each day.

If you select periodic meters, you must specify an offset from midnight (the base) as well as an interval period. The EGM uses the base plus the interval to determine the first periodic trigger. The defined interval period is then used to determine subsequent periodic triggers. For example, if:

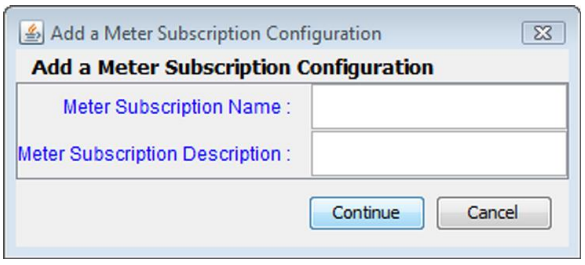
- Periodic Base = 60:00
- Periodic Interval = 15:00

then, the first periodic meter report will be sent by the EGM at 1:15am. Subsequent periodic reports will be sent by the EGM at 1:30am, 1:45am, 2:00am and so on, until the periodic interval is changed.

Add or Edit a Meter Subscription

You can add a new meter subscription or edit an existing meter subscription through the Meter Subscription option on the Components tab.

1. Go to: **Custom Configuration** *or* **Tester Toolkit** > **Components** > **Meter Subscriptions**
2. To create a new meter subscription:
   a. Click **Add** to add a new meter subscription.



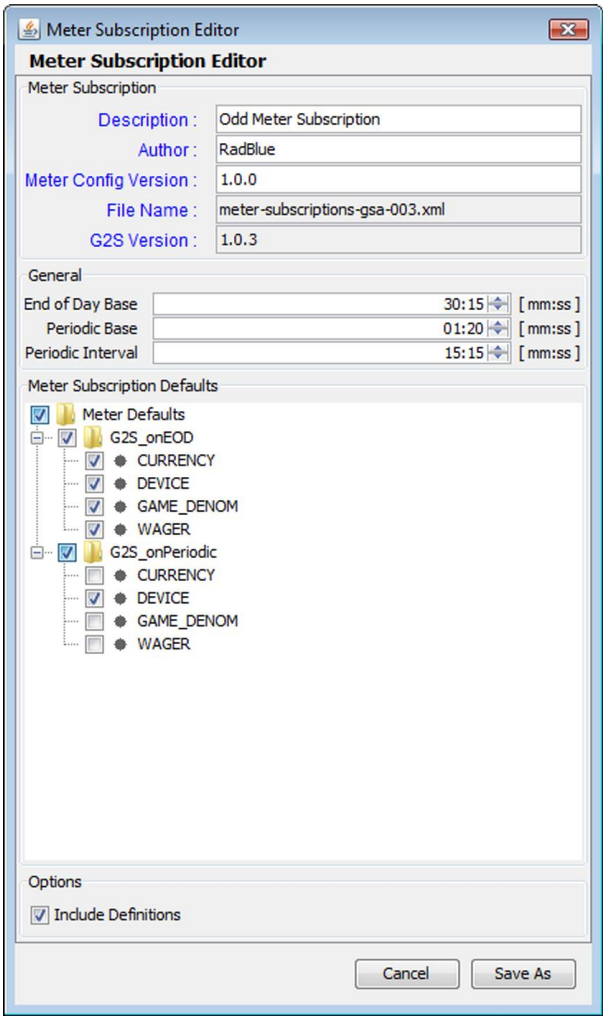
   b. Type a name and description for the new meter subscription.
   c. Click **Continue**.

   or

   To open an existing meter subscription to edit:

a. From **Available Meter Subscriptions**, click to select the meter subscription you want to edit.

3. Click **Edit**.



4. Modify information for the meter subscription configuration. Note that RGS automatically inserts default information.
   - **Description** - Type a description for the meter subscription.
   - **Author** - Type an identifier for the meter subscription author.
   - **Meter Config Version** - Type a version for number for the meter subscription. This field can contain any information required.
   - **File Name** - File name of the meter subscription. This field is *read-only*.

- **G2S Version** - Indicates the selected version of the Game To System (G2S) protocol. The G2S version is defined in the Schema Option field under **Configure** > **Engine Options**. This field is *read-only*.

5. Configure the meter subscription.

   General

   - **End of Day Base** - Type the amount of time (minutes:seconds) after midnight that end-of-day meters are sent each day.
   - **Periodic Base** - Type the amount of time (minutes:seconds) after midnight that is used to establish the periodic trigger.
   - **Periodic Interval** - Type the frequency (minutes:seconds) that periodic meters are sent by the EGM.

   Meter Subscription Defaults

   - **ON_EOD** - Select ON_EOD to subscribe to the following meters: currency, device, game_denom, and wager. If you want to subscribe only to specific end-of-day meters, select the appropriate meter(s).
   - **ON_PERIODIC** - Select ON_PERIODIC to subscribe to the following meters: currency, device, game_denom, and wager. If you want to subscribe only to specific periodic meters, select the appropriate meter(s).

   Options

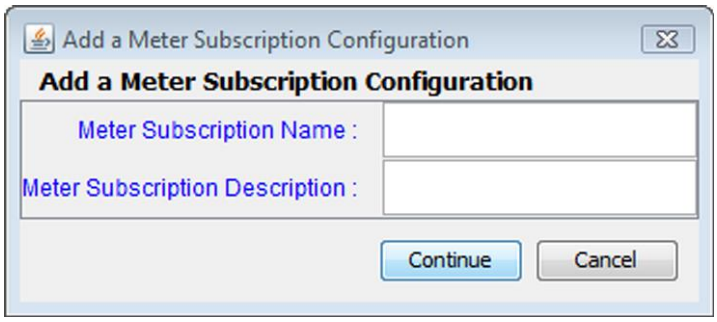
   - **Include Definitions** - Select to include meter definitions in meter reports.

6. If you are adding a new meter subscription, click **Save**.

   or

   If you are editing an existing meter:

   a. Click **Save As**.



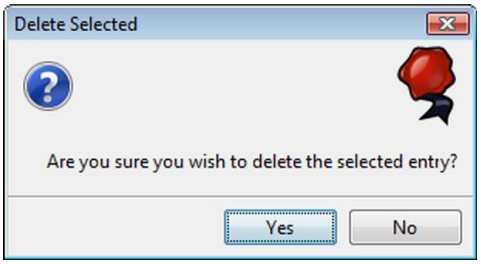
   b. Type a name and description for the new meter subscription.

   c. Click **Continue**.

Delete a Meter Subscription

The Remove option lets you delete saved custom meter subscriptions as needed. Note that pre-configured meter subscriptions, denoted by a lock symbol to the left of the entry, cannot be deleted.

1. Click to select the meter subscription you want to delete.
2. Click **Remove**.

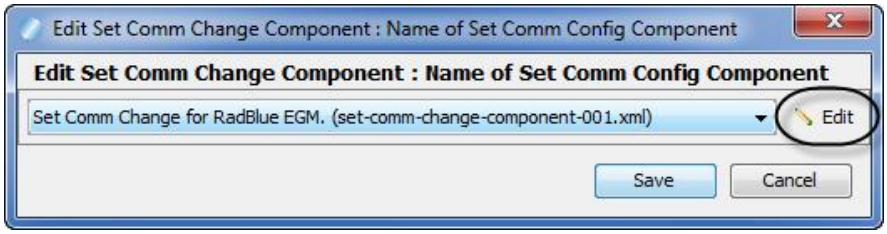


3. Click **Yes** to delete the selected meter subscription.

Using Set Comm Change File

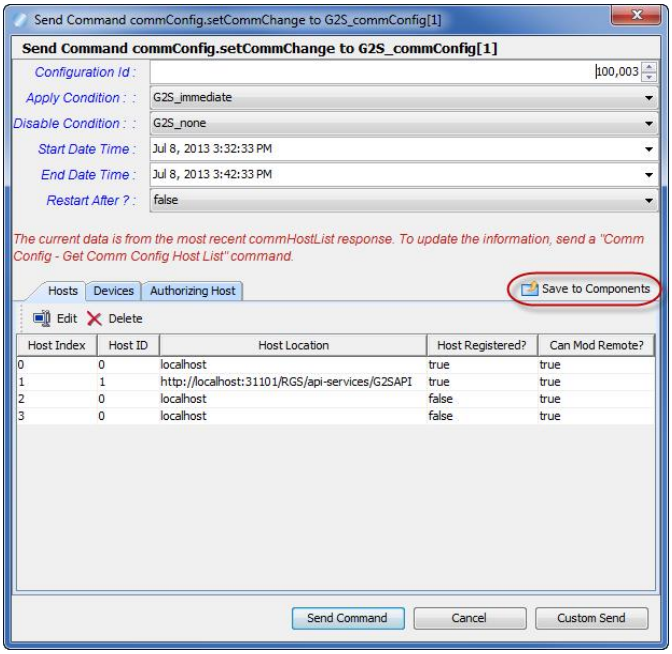
Use this procedure to save a `commConfig.setCommChange` command configuration to the Available Set Comm Change Files list on the Set Comm Change component. You can also edit or delete a Set Comm Change file.

Once saved to the list, the file is available for use in custom scripts. You can edit a configuration from the Custom Scripting interface when you add the `commConfig.setCommChange` command to a script.

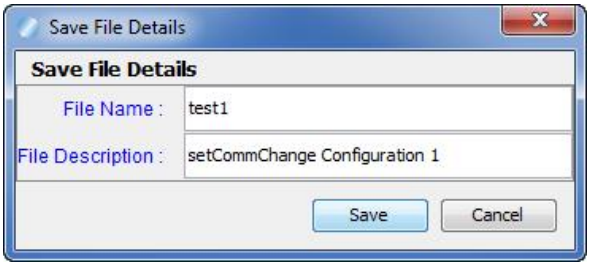


To save a `commConfig.setCommChange` command configuration to the Available Set Comm Change Files list:

1. Go to: **Send Command > G2S_commConfig**
2. Under Available Commands, highlight **setCommChange**, and click **Send Selected Command** (or double-click the command).



3. Configure the `commConfig.setCommChange` command as required.
4. Click **Save to Components**.



5. Type the **File Name** and **File Description**, and click **Save**.
6. Click **Cancel** or **Send Command**.
7. Go to: **Tester Toolkit** > **Components** > **Set Comm Change**
8. Verify that the file has been added to the **Available Set Comm Change Files** list.

Edit a Set Comm Change File

Once you save a `commConfig.setCommChange` command to the Set Comm Change component, it is ready to be used in any custom script in the Tester Toolkit. However, you have the option of editing the `commConfig.setCommChange` command as needed.

> **Note:** Any changes you make to the **`commConfig.setCommChange`** command's XML must be valid.

1. Go to: **Tester Toolkit** > **Components** > **Set Comm Change**
2. From the **Available Set Comm Change Files** list, click to select the component you want to edit.

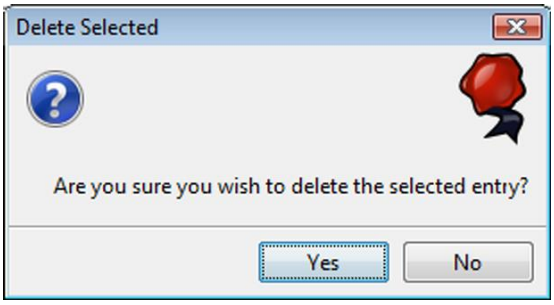
   Note: If you do not see the component you want to edit in the list, click **Refresh**.
3. Click **Edit**.
4. Modify the component information as needed.
   - **Description** - Component description.
   - **Author** - Component author.
   - **Version** - Component version.
   - **File Name** - Name of file.
   - **G2S Version** - G2S version to be used with component.

   Note: The unique configuration ID value is added automatically when the custom script runs.
5. Modify the `commConfig.setCommChange` command's XML as needed. **All XML changes must be valid**.
6. Click **Save**.

Delete a Set Comm Change File

Use this procedure to remove a file from the Available Set Comm Change Files list.

1. Go to: **Tester Toolkit** > **Components** > **Set Comm Change**
2. Click to highlight the file you want to delete.
3. Click **Remove**.

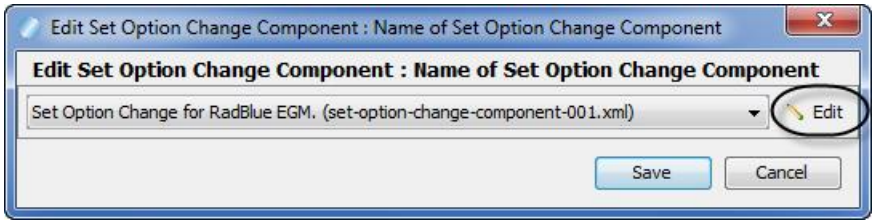


4. Click **Yes** to delete the selected file, or click **No** to return to the Set Comm Change screen without deleting the selected file.

Using Set Option Change

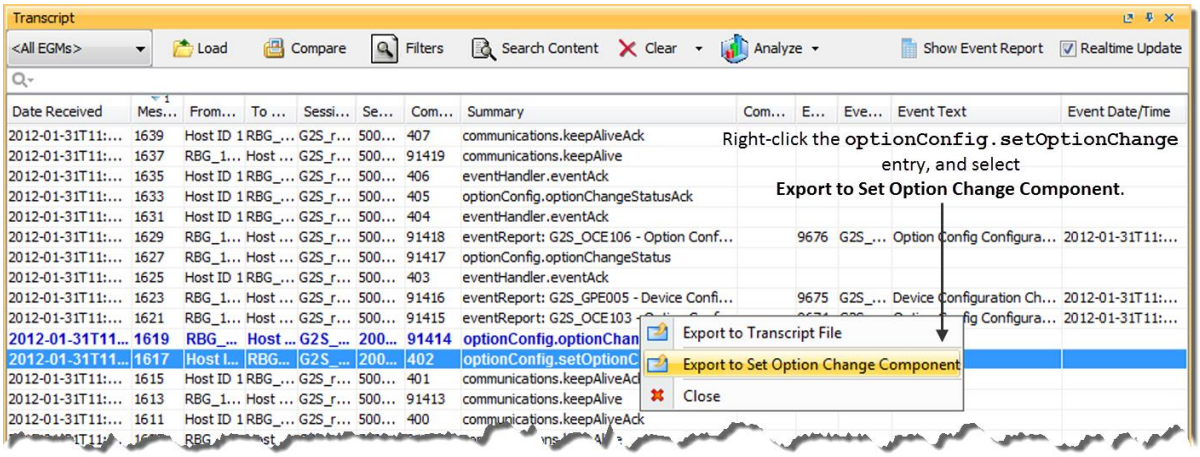
The Set Option Change component lets you edit `optionConfig.setOptionChange` commands, and include them in custom scripts.

You can edit a Set Option Change component from the Custom Scripting interface when you add the `optionConfig.setOptionChagne` command to a script.

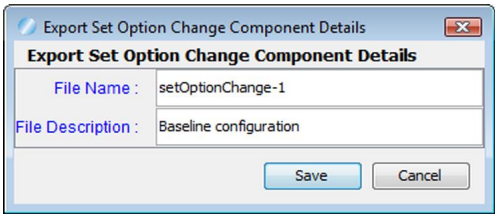


To use the Set Option Change component:

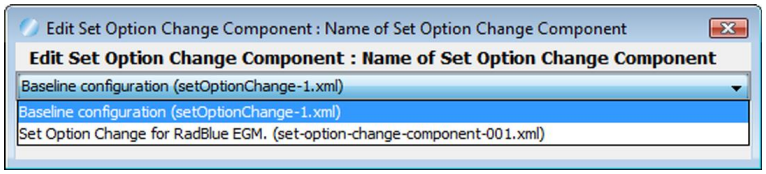
1. Configure an `optionConfig.setOptionChange` command through the Send Command.
2. Send the `optionConfig.setOptionChange` command to the EGM.
3. Look at the Transcript.



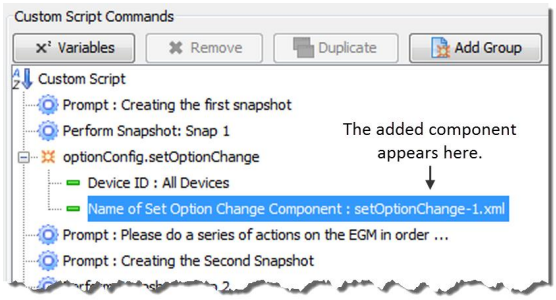
4. Right-click the **setOptionChange** entry, and select **Export to Set Option Change Component**.



5. Type a name and description for the `setOptionChange` component.
6. Click **Save** to add the `setOptionChange` command to the **Available Set Option Change Files** list.
7. Confirm that the correct component was added, and click **OK**.
8. Go to: **Tester Toolkit** > **Components** > **Set Option Change**
9. Click **Refresh** to refresh the **Available Set Option Change Files** list. The new component should appear in the list.
10. Edit the component as needed.
11. To use the component in a custom script:
    a. Go to **Tester Toolkit** > **Custom Scripting**
    b. Select a script.
    c. From the **Available Commands** list, drag-and-drop **optionConfig.setOptionChange** into the **Custom Script Commands** list.
    d. Click the plus symbol (+) in front of the **optionConfig.setOptionChange** entry to expand it.
    e. Double-click **Name of Set Option Change Component**.



    f. Click the drop-down arrow, and select the component you want to insert into the script.
    g. Click **Save**. Your selection is reflected in the `setOptionChange` command options:



Edit a Set Option Change File

When you export an `optionConfig.setOptionChange` command into the Set Option Change component, it is ready to be used in any custom script from i Tester Toolkit, an optional RGS module. However, you have the option of editing the `optionConfig.setOptionChange` command as needed.

Note: Any changes you make to the `optionConfig.setOptionChange` command's XML must be valid.

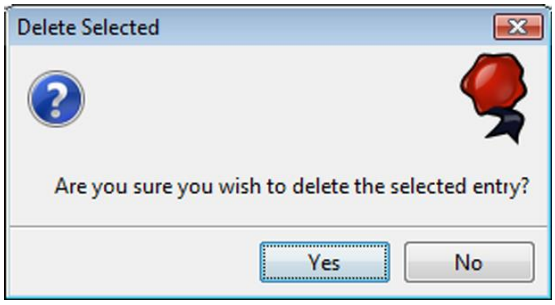
1. Go to: **Tester Toolkit** > **Components** > **Set Option Change**
2. From the **Available Set Option Change Files** list, click to select the component you want to edit.

   Note: If you do not see the component you want to edit in the list, click **Refresh**.
3. Click **Edit**.
4. Modify the component information as needed.
   - **Description** - Component description.
   - **Author** - Component author.
   - **Version** - Component version.
   - **File Name** - Name of file.
   - **G2S Version** - G2S version to be used with component.
5. Modify the `optionConfig.setOptionChange` command's XML as needed. **All XML changes must be valid.**
6. Click **Save**.

Delete a Set Option Change File

Use this procedure to remove a file from the Available Set Option Change Files list.

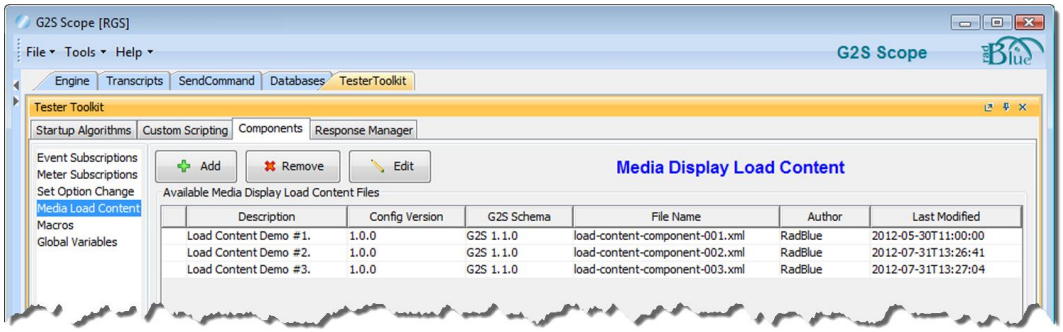
1. Go to: **Custom Configuration** > **Components** > **Set Option Change**
2. Click to highlight the file you want to delete.
3. Click **Remove**.



4. Click **Yes** to delete the selected file, or click **No** to return to the Set Option Change screen without deleting the selected file.

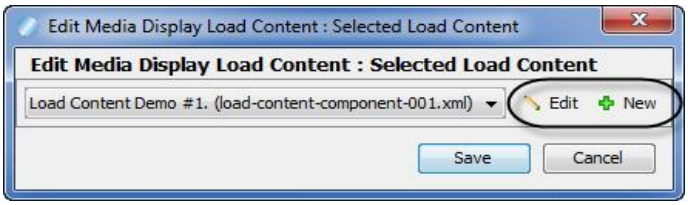
Using Media Load Content

The Media Load Content component lets you define media content for use with the `mediaDisplay.loadContent` command in Custom Scripting. You can add or edit media display files.



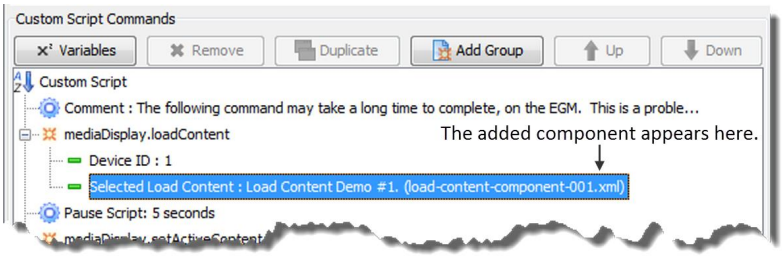
To use the component in a custom script:

1. Go to **Tester Toolkit** > **Custom Scripting**
2. Add a new script or edit an existing script.
3. From the **Available Commands** list, drag-and-drop **mediaDisplay.loadContent** into the **Custom Script Commands** list.
4. Click the plus symbol (+) in front of the **mediaDisplay.loadContent** entry to expand it.
5. Double-click **Selected Load Content**.



Note: From the Edit Media Display Load Content screen, you can add or edit a `media.loadContent` component by clicking either **Edit** or **New**.

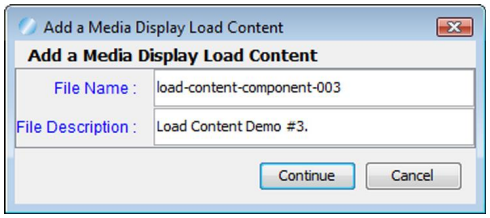
6. Click the drop-down arrow, and select the media display file you want to insert into the script.
7. Click **Save**. Your selection is reflected in the `mediaDisplay.loadContent` command options:



Add or Edit a Media Display File

You can add or edit media display files through the Media Load Content option on the Components tab. Files in the Available Media Display Load Content Files list are automatically made available for the `mediaDisplay.loadContent` command when you create a custom script.

1. Go to: **Tester Toolkit** > **Components** > **Media Load Content**
2. To create a new media display file:
   a. Click **Add**.

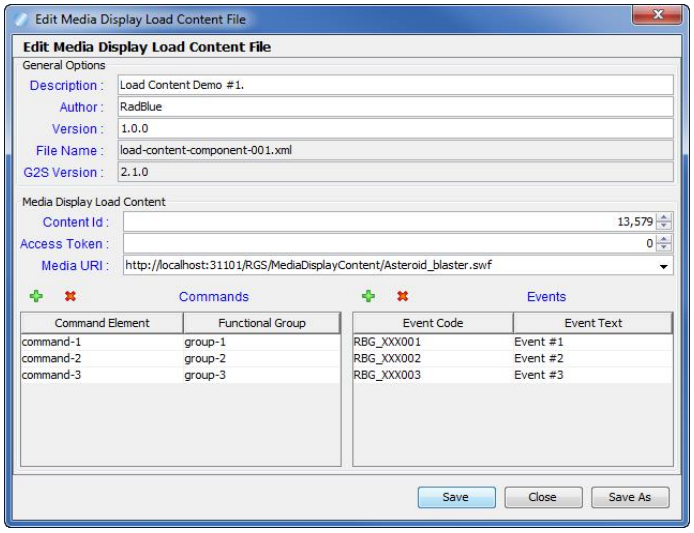


   b. Type a file name for the media display file (spaces are not allowed). Note that the file name must be unique.
   c. Type a description for the media display file.
   d. Click **Continue**.

   or

   To open an existing algorithm to edit:

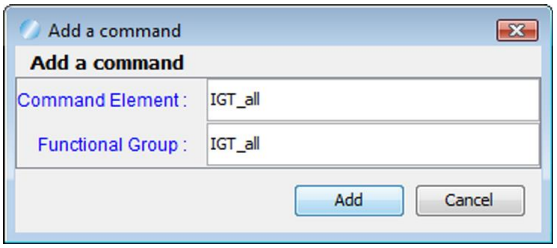
   a. Click to highlight the algorithm you want to edit.
   b. Click **Edit**.



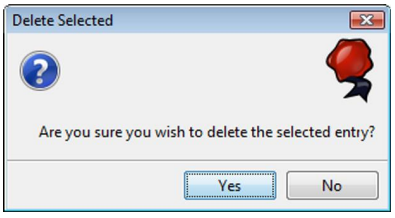
3. Configure the media display file information as needed.

- **Description** - Type a description for the media display file.
- **Author** - Type an identifier for the media display file's author.
- **Version** - Type a version number for the media display file. This field can contain any information required.
- **File Name** - File name of the media display file. This field is *read-only*.
- **G2S Version** - Indicates the selected version of the Game To System (G2S) protocol. The G2S version is defined in the Schema Option field under **Configure** > **Engine Options**. This field is *read-only*.

4. Configure the media display load content file information as needed.
   - **Content ID** - Number used by the host to refer to this specific piece of content. Note that the Content ID value must be the same as the Content ID value in the `mediaDisplay.setActive` and `mediaDisplay.releaseActiveContent` commands.
   - **Media URI** - Location from which the EGM will download this piece of content.
3. Add commands supported by the EGM is authorized to execute when the media display file content is loaded.
   a. Click the plus (+) icon under the Commands heading to add a supported command.

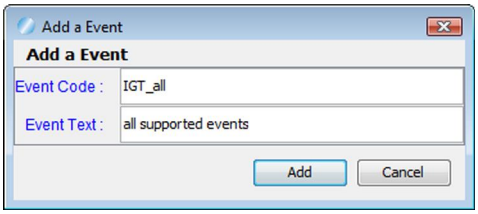


   a. Type the command element and functional group.
   b. Click **Add**. The new command appears in the Commands list.
   c. To delete an existing command, click to select a command in the Command list, and click the **Remove** icon.

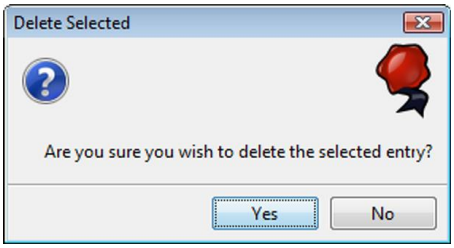


   d. Click **Yes** to delete the selected command.

6. Add events the EGM is authorized to recieve when the media display file content is loaded.
   a. Click the plus (+) icon under the Events heading to add a supported command.



   a. Type the the event code and descriptive text.
   b. Click **Add**. The new event appears in the Events list.
   c. To delete an existing event, click to select an event in the Events list, and click the **Remove** icon.



   Click **Yes** to delete the selected command.

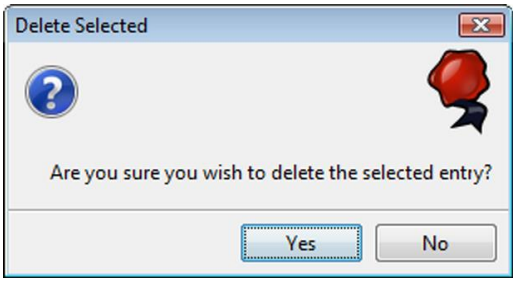
7. Click **Save** or **Save As**, as required, to save the media display file. All saved files display in the Available Media Display Load Content list.

   If you choose **Save As**, remember that the file name must be unique.

Delete a Media Display File

Delete any media display file from the Available Media Display Load Content Files list. Once a media display file is deleted, it is no longer available for use in Custom Scripting.

1. Go to: **Tester Toolkit** > **Components** > **Media Load Content**
2. From the **Available Media Display Load Content Files** list, click to select the file you want to delete.
3. Click **Remove**.



4. Click **Yes** to delete the selected file.

Using Macros

Macros let you create pre-configured, multi-command host operation scripts (for example, the downloading and installation of a package). Each macro that you create automatically becomes available in Custom Scripting. You can add or edit a macro directly from the Custom Scripting interface by adding the **Run Macro** script verb to your script, and double-clicking **Select Macro to Run**.

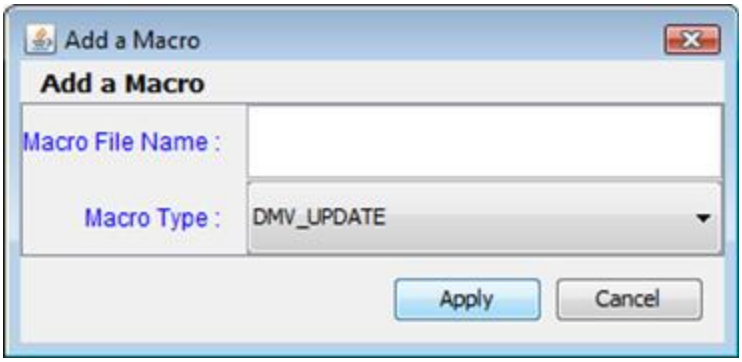
Once you create a macro, you can change its settings or delete it at any time. All default and newly created macros are saved to the following location:

[**RGS installation directory**] > **custom-config** > **script-runner**

Add a Macro

This procedure lets you add a new script to Custom Scripting. This script can then be run as many times as necessary with the click of a button.

1. Go to: **Custom Configuration** > **Components** > **Macros**
2. Click **Add**.



3. Type a name for the macro (spaces are not allowed). Note that the macro file name must be unique.
4. Click the drop-down arrow, and select the type of macro you want to create.
5. Click **Apply**.
6. Configure the macro.

   For information on configuring the **Download a Package** macro, see Configure the Download a Package macro.

Edit a Macro

1. Go to: **Custom Configuration** > **Components** > **Macros**
2. Click to select the macro you want to edit.
3. Click **Edit**.
4. Modify the macro configuration.
   For information on configuring the **Download a Package** script, see Configure the Download a Package macro.

Configure the DMV Update Macro

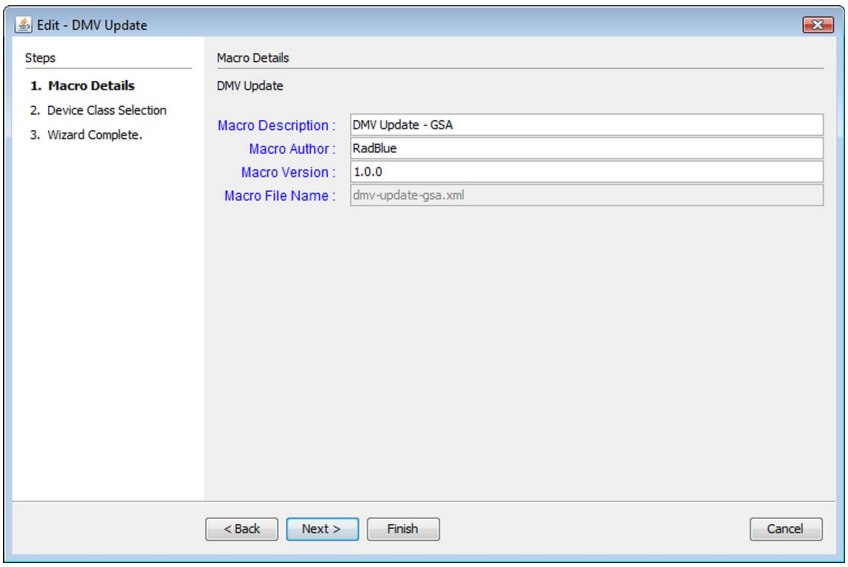
The DMV Update macro lets you request the latest data model information for a specific EGM. This macro requests the status, meters, profile and option list for every device in the selected class.

Note: Macros are available only with the optional Tester Toolkit module.

1. Go to: **Custom Configuration** > **Components** > **Macros**
2. Click **Add** and create a new macro. See Add a Macro for more information.

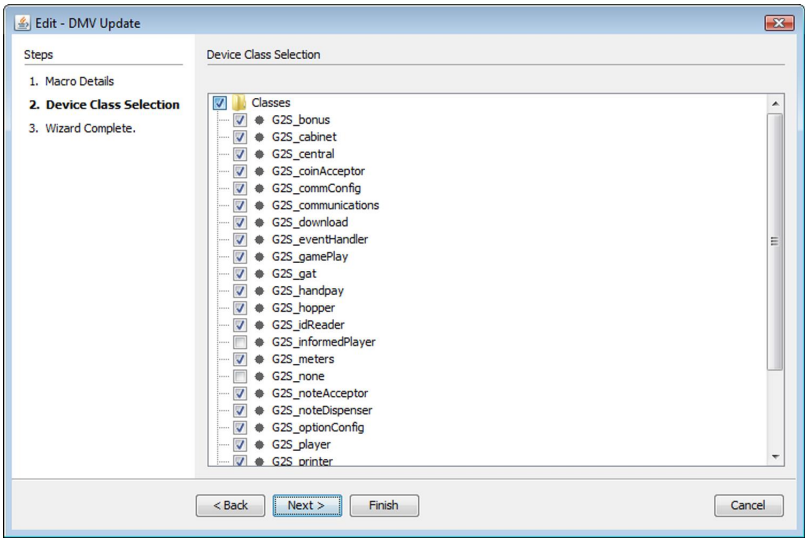
   or

   Click to select the download macro you want to edit, and click **Edit**.

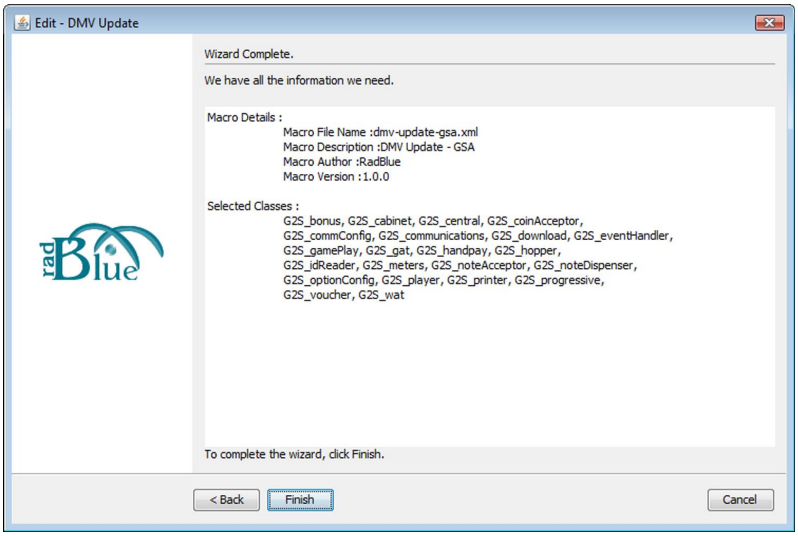


3. Modify identifying information for the algorithm. Note that RGS automatically inserts default information.
   - **Macro Description** - Type a description for the macro.
   - **Macro Author** - Type an identifier for the macro's author.
   - **Macro Version** - Type a version for number for the macro. This field can contain any information required.
   - **Macro File Name** - File name of the macro. This field is *read-only*.

4. Click **Next**.



5. Select the classes for which you want updated data model information.
6. Click **Next**.



7. Review the macro information.
8. Click **Back** to make additional changes to the macro, click **Finish** to save the macro or click **Cancel** to exit the wizard without saving your changes.

Configure the Download a Package Macro

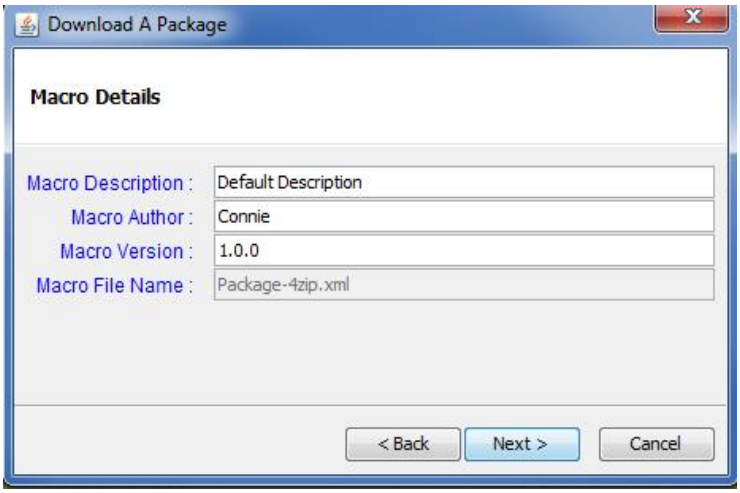
The Download a Package macro lets you configure the options to download a package to an EGM. You can define the transfer location, transfer type and package identifier. You can choose to authenticate the

package, define package authorizers and install the package once it's downloaded.

Note: Macros are only available with the optional Tester Toolkit module.

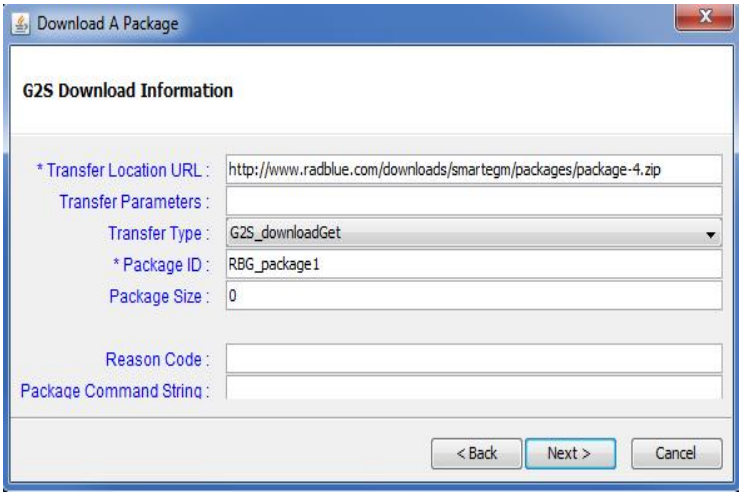
1. Go to: **Tester Toolkit** > **Components** > **Macros**
2. Click **Add** to create a new macro. See Add a Macro for more information.
   or

   Click to select a download macro you want to edit, and click **Edit**.



3. Modify identifying information for the algorithm. Note that RGS automatically inserts default information.
   - **Macro Description** - Type a description for the macro.
   - **Macro Author** - Type an identifier for the macro author.
   - **Macro Version** - Type a version for number for the macro. This field can contain any information required.
   - **Macro File Name** - Package file name. This field is read-only.

4. Click **Next**.

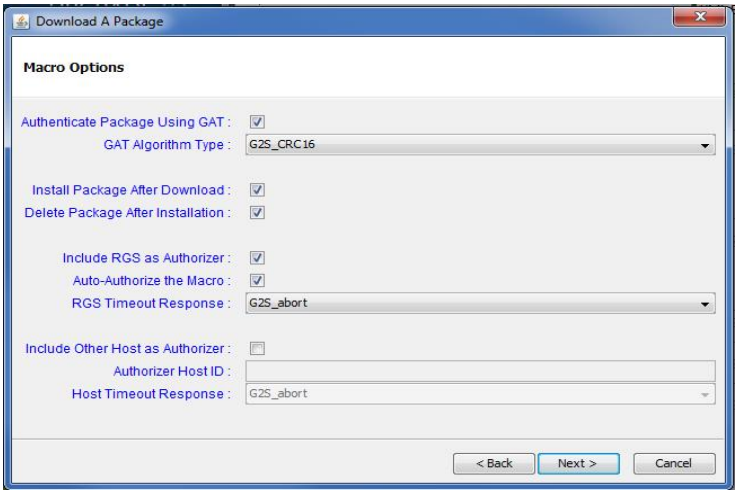


5. Complete the information for the download. Required fields are denoted with an asterisk (*).
   - **Transfer Location URL** - Enter the network address of the location on the Software Download Distribution Point (SDDP) where the package resides.
   - **Transfer Parameters** - Enter any parameters required for the package transfer. This field is optional.
   - **Transfer Type** - Select whether the host or EGM initiates the transfer as well as the direction of the transfer.
   - **Package ID** - Type a unique identifier for this package on the EGM. The first three characters must be a GSA-assigned manufacturer identifier, followed by an underscore character ("_").

   Note: If you install a package more than once, the package ID *must* be unique each time you add the package. Otherwise, a `G2S_DLX002` error will be generated when the Set Script for Package command is sent.

   - **Package Size** - Size of package according to the System Management Point (SMP). A value of zero (0) indicates that the file size is unknown. This field is optional.
   - **Reason Code** - Enter the reason for the transfer. This field is optional.
   - **Package Command String** - Enter script command elements that you want to execute with the package. The EGM validates command strings prior to starting the macros. This field is optional.

6. Click **Next**.



7. Set the options for the macro.
   - **Authenticate Package Using GAT** - Select to authenticate the package on the EGM, once it is downloaded, using the Game Authentication Terminal (GAT) class.
   - **GAT Algorithm** - Select the authentication algorithm you want the EGM to use to authenticate the package.
   - **Install Package After Installation** - Select to automatically install the package when download is complete.
   - **Include RGS as Authorizer** - Select if you want RGS to authorize the macro.
   - **Auto-Authorize the Macro** - Select if you want RGS to automatically authorize the macro. If this option is cleared, the macro waits until you manually authorize the package.

     To manually authorize a package, go to:

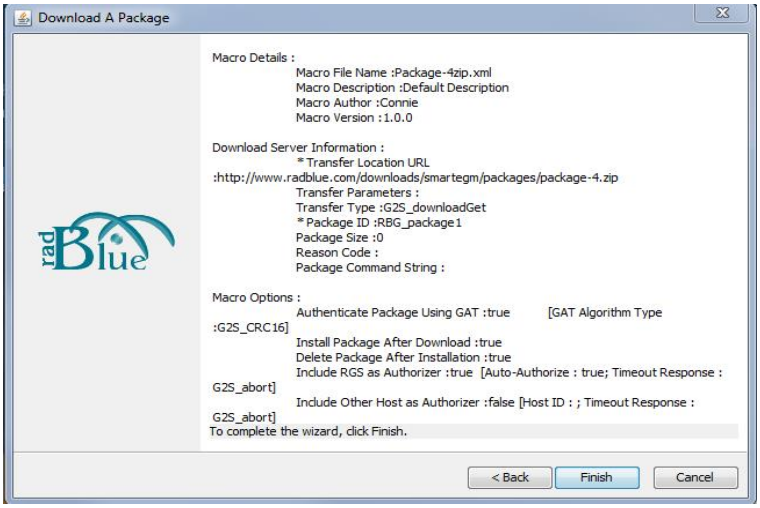
     Send Command > **G2S_download** > **authorizeScript**

     i. The **Script ID** is automatically updated to reflect your macro's identifier.

     ii. Click the **Authorized** drop-down arrow, and select **true** to authorize the macro.

     iii. Click **Send Command**.
   - **RGS Timeout Response** - Select to continue the operation if the host authorization times out (`G2S_ignore`) or to cancel the download installation script if the host authorization times out (`G2S_abort`).
   - **Include Other Host as Authorizer** - Select if you want to include a host other than RGS as an authorizing host.
   - **Authorizer Host ID** - Type the host identifier of the authorizing host.
   - **Host Timeout Response** - Select to continue the operation if the host authorization times out (`G2S_ignore`) or
     to cancel the download if the host authorization times out (`G2S_abort`).

8. Click **Next**.

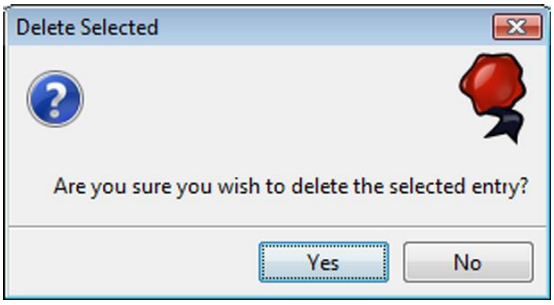


9. Review the macro information, and click **Finish**.
   The new macro is added to the **Available Macros** list.

Delete a Macro

Use this procedure to remove a script from the Available Macros list.

1. Go to: **Custom Configuration** > **Components** > **Macros**
2. Click to highlight the macro you want to delete.
3. Click **Remove**.



4. Click **Yes** to delete the selected macro, or click **No** to return to the Macros screen without deleting the selected macro.

Using Global Device Variable Templates

When you add a command to a custom script, you have the option to select the device you want the command to go to. By default, each command goes to all devices; however, you can use global variable templates to globally change device settings. Once global variables are defined through Global Variables option, a copy of the variable template becomes available in Custom Scripting. You can then modify the local copy as needed.

There are three standard variable templates:

- **All Devices** - Command is sent to all devices in the device class.
- **First Device** - Command is sent only to the first device in the device class.
- **Pick a Device** - Lets you select a device when the custom script is run.

Standard variables cannot be modified. You can create custom global variables that you can modify as needed through the Global Variables option.

Finally, you can create script-specific variables using the Variables option when you add or edit a custom script. See Add a Device Variable to a Custom Script.

Add or Edit a Global Device Variable Template

ou can add and edit custom global device variable templates through the Global Variables option on the Components screen. New device variables can only be used in custom scripts created after the variable.

1. Go to: **Tester Toolkit** > **Components** > **Global Variables**
2. To add a variable:
   a. Click **Add**.

   b. Configure the device template.
      - **Name** - Type a name for the device variable template.
      - **Device Class** - Click the drop-down arrow, and select the device class you want to associate with the device variable.
      - **Type** - Click the drop-down arrow, and select the device type.
      - **Description** - Type a description for the device variable template.
      - **Default Value Selection** - Click the drop-down arrow, and select a device or action for the variable. In addition to All Devices, Pick a Device and First Device, you can choose to always use a specific device identifier (**Hard Coded Device ID**).

   or

   To edit a variable:

   a. Click to highlight the variable you want to edit.
   b. Click **Edit**.
   c. Modify the variable as needed.
3. Click **Save**.

Delete a Global Device Variable Template

You can delete a custom device variable template at any time. Device variables based on the template you are deleting are not affected by the deletion of a template.

Standard device variable templates cannot be deleted.

1. Go to: **Tester Toolkit** > **Components** > **Global Variables**
2. Click to highlight the variable you want to delete.
3. Click **Delete**.

About Custom Scripting

With Custom Scripting, you can create powerful scripts of any G2S host-initiated commands through an easy-to-use graphical interface. Custom Scripting can be used by test engineers to develop complex scripts that can then be run over and over by testers.

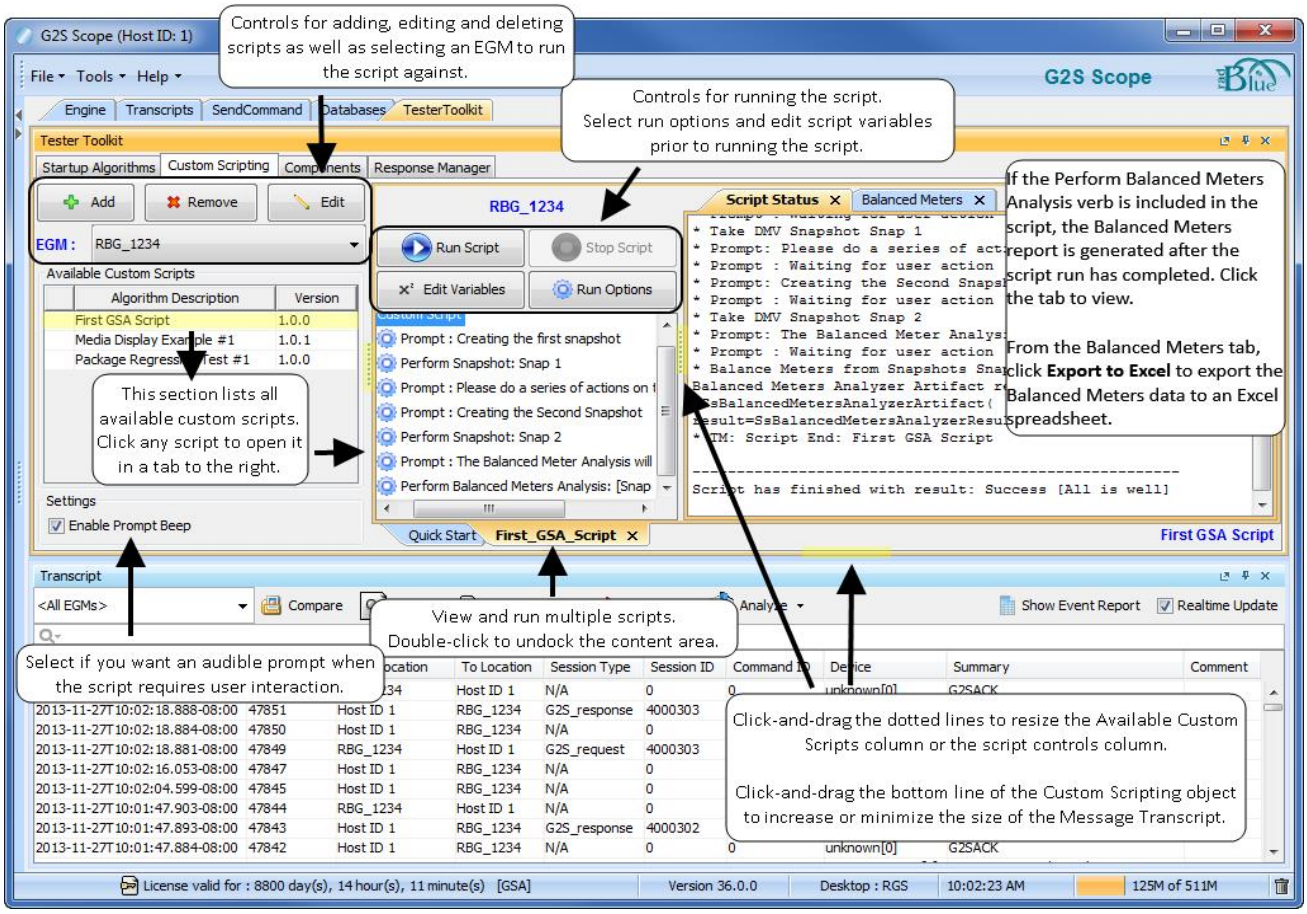
In addition to being able to send any G2S command, scripts can include instructions to testers as well as prompts (for example, “Insert a $5 bill, and then press Enter”). Scripts can even validate that the EGM meter model is still in balance at selected points in the script.

You can not run more than one custom script at a time.

The GAT Management Control option lets you create digest files that can then be used to verify selected components in other EGMs against the results from the known-good software on the control EGM.

Review the Custom Scripting Layout

The Custom Scripting tab is located on the Tester Toolkit layout and contains options to create and manage custom scripts as well as the Message Transcript object.

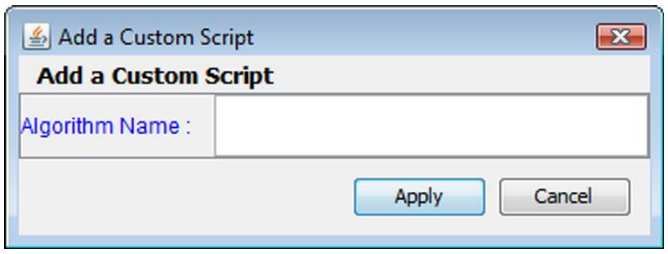


Working with Custom Scripts

Add or Edit a Script

When you create a custom script, you can either start with an empty script, or you can use an existing script that you save with a different file name.

1. Click the **Custom Scripting** tab on the Tester Toolkit layout.
2. To create a new custom script:
   a. Click **Add**.



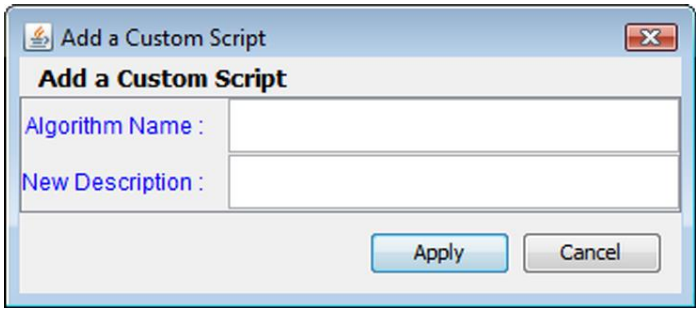
b. Type a name for the script, and click **Apply**.

or

a. Click to select a script from the **Available Custom Scripts** list.

b. Click **Edit**.

c. Once you have finished editing the script, click **Save As**.



d. Type a new name and description for the script.

e. Click **Apply** to save the new script.

Build a Custom Script

Custom Scripting supports most of the G2S host commands as well as script verbs, which include complex commands and script-specific functions. Building a custom script is as easy as moving commands and script verbs into the Custom Script Commands list. Once added, you can modify command attributes and verb settings to further customize the script.

The **Custom Script Information** section is auto-populated. Modify the default values as needed.

Build a custom script by moving verbs and commands, which appear in the **Script Verbs** and **Available Commands** section on the right of the screen, into the **Custom Script Commands** section on the left side of the screen.

- To move a verb or command into the Custom Script Command section, click to highlight it and:
  - click the left arrow
  - double-click
  - drag and drop

  You can CTRL+click to select multiple commands.
- To add a command to the end of the script, highlight the command and click .
- When you move a Script Verb into a custom script, you are prompted to configure that verb. See About Script Verbs.
- Commands with attributes have a plus symbol (+) in front of the command that you can expand to view all configurable attributes. To configure an attribute, double-click it, modify the attribute value as needed, and click Save.
- After you have configured a command's attributes, you can create a copy of that command by highlighting it and clicking **Duplicate**.
- To move commands and verbs within the script:
  - drag and drop
  - click the up and down arrows
- To create a group of commands, click to highlight the **Custom Script** text at the top of the tree, and click **Add Group**. Click the drop-down arrow, and select the device class for the command group you are creating.
- Move the commands you want included in the group from the **Available Commands** list, and configure each command as needed. A command group can only contain commands from the selected device class.
- One G2S message containing all of the selected commands will be sent to the EGM for every device within that class.

- To delete a command, command group or verb, click to highlight the content you want to delete, and click **Remove**. Use CTRL+click to highlight multiple commands.
- To add a copy of a command, command group or verb to the script, including configuration settings, click to highlight the content you want to copy, and click **Duplicate**.
- Click **Save** (or **Save As**, if you're creating a new script from an existing one) to save your changes.

Add a Device Variable to a Custom Script

When you add a command to a custom script, you have the option to select the device you want the command to go to. By default, each command goes to all devices within the relevant class; however, you can use device variables to customize your script so the command is sent to a single device, hard-coded or selected at run-time. For convenience, a set of global variable templates are provided that let you globally change device settings as well as the default device setting. Once global variable templates are defined through Global Variables option on the Components tab, a copy of the variable template becomes available in any new scripts that you create using Custom Scripting. You can then modify the local copy as needed.

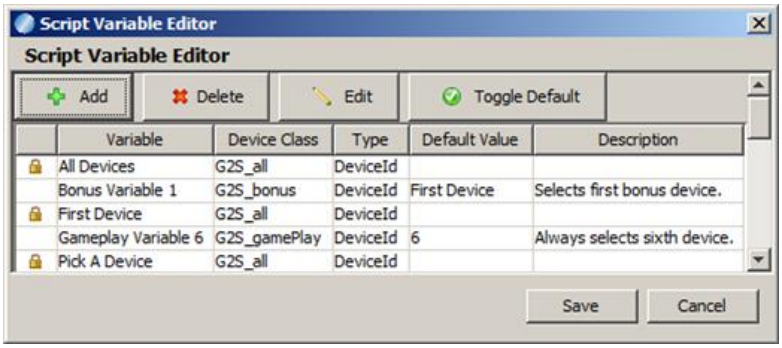
There are three standard device variables:

- **All Devices** - Command is sent to all devices in the device class.
- **First Device** - Command is sent only to the first device in the device class.
- **Pick a Device** - Lets you select a device when the custom script is run.

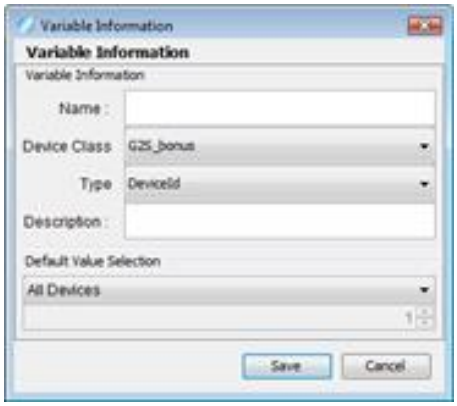
Standard variables cannot be modified. You can create your own global variables that you can modify as needed through the Global Variables option. See Using Global Device Variable Templates.

Finally, you can create script-specific variables using the Variables option when you add or edit a custom script. See Add a Device Variable to a Custom Script.

1. Go to: **Tester Toolkit** > **Custom Scripting**
2. Add or edit a custom script.
3. Click **Variables**.



4. To add a device variable:
   a. Click **Add**.



   b. Configure the device template.
      - **Name** - Type the name of this variable.
      - **Device Class** - Click the drop-down arrow, and select the device class for this device variable.
      - **Type** - Click the drop-down arrow, and select the ID type.
      - **Description** - Type a description for this variable.
      - **Default Value Selection** - Click the drop-down arrow, and select a device or action for the variable. In addition to All Devices, Pick a Device and First Device, you can choose to always use a specific device identifier (Hard Coded Device ID).
   c. Click **Save**.
5. To edit a device variable:
   a. Click to highlight the variable you want to edit.
   b. Click **Edit**.
   c. Modify the variable as needed.
   d. Click **Save**.
6. To delete a device variable:
   a. Click to highlight the variable you want to delete.
   b. Click **Delete**.
7. To change the default device variable, click to highlight the variable you want to make the default, and click **Toggle Default**. Note that you can specify a default variable for each device class.
8. Click **Save** to retain your changes.

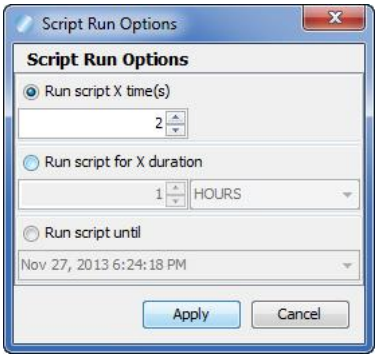
Run a Custom Script

You can easily run any available script from the Custom Scripting object. Prior to running a script, you can choose the EGM you want to run the script against, how many times or how long you want the script to run, and edit device variable settings.

1. From the Tester Toolkit layout, click **Custom Scripting**.
2. Click any script in the **Available Custom Script** list.

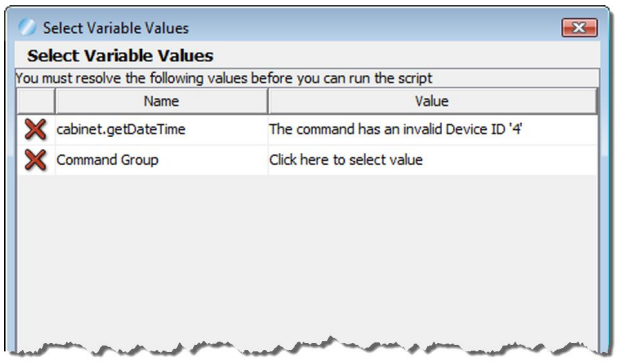
   A tab for the selected script opens to the right of the list. This tab includes the Run Script option, an option to edit the script device variables, an option to define how you want the script to run and a list of all commands (or command groups) in the script. An option to close the tab is not available until the tab is selected.

3. Click the **EGM** drop-down arrow above the Available Custom Scripts list, and select the EGM you want to run the script against. The selected EGM displays in the script tab.
4. Click **Run Options** to run the script multiple times.

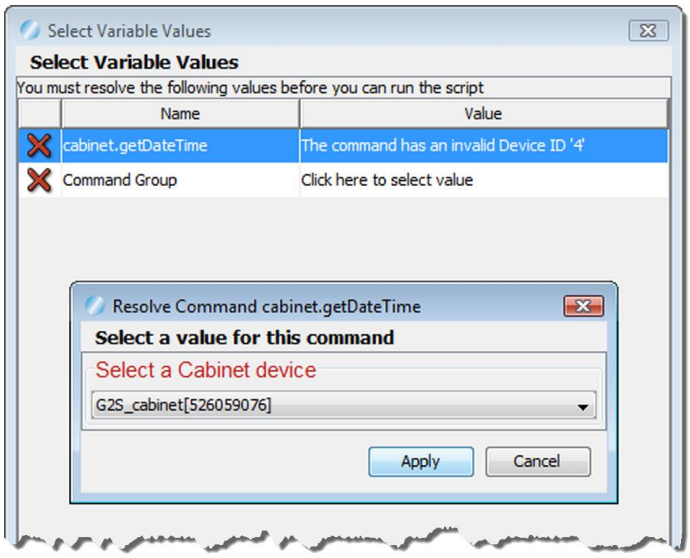


5. Select one of the following options:
   - **Run script x time(s)** - Type the number of times you want the script to run. By default, this option is selected and set to **1**.
   - **Run script for X duration** - Select how long to run the script (for example, **3 hours**).
   - **Run script until** - Select the date and time you want the script to stop running.

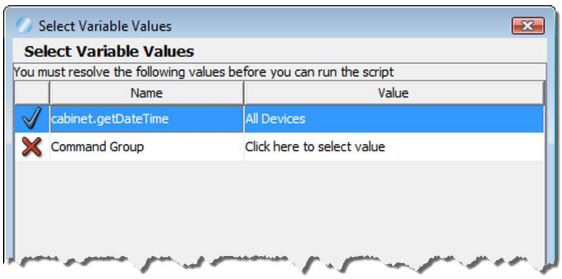
6. Click **Apply**.
7. Click **Run Script**.
   If you have device or command variables that need to be resolved, they are displayed to you.



5. Double-click a row to resolve the variable.



6. Click the drop-down arrow, and select a device for the command.
7. Click **Apply**.
8. Once a variable has been resolved, and check mark displays in front of the row.



9. As soon as all variables have been resolved, click **Apply** to continue running the script.

   When a script runs, each command in the script is highlighted in the command list and the Script Status displays each command as it executes. Any user input prompts are displayed under Script Status. Finally, a green bar under the Run Script button shows you the script's progress.

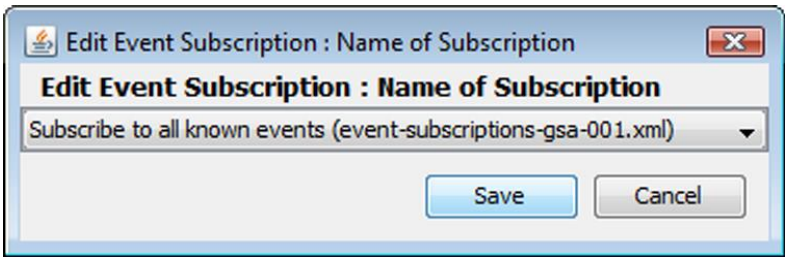
   If you included the Perform Balanced Meters Analysis verb in your script, a **Balanced Meters** tab displays next to the Script Status tab that displays the results of that analysis.

   Note that you can detach a script tab from the main application screen to view and run multiple scripts at once. Double-click the tab to undock it, and close the script window to move it back into the main application.

Insert Event Subscription File into a Custom Script

You can add an event subscription file that you created in the Components section to a custom script simply by inserting the setEventSub command into the script.

1. Go to: **Tester Toolkit** > **Custom Scripting**
2. Click **Add** to create a new script, or select an existing script and click **Edit**.
3. Scroll to **eventHandler.setEventSub** in the Available Commands list.
4. Left-click and drag the command into the Custom Script Commands section.
5. Click the plus sign to expand the command information.
6. Double-click **Name of Subscription**.



7. Click the drop-down arrow, and select the event subscription file you want to run at the point you have inserted the `setEventSub` command in the script.

8. Click **Save**.
9. Click **Save** at the bottom of the **Edit a Custom Script** screen to save your changes and exit back to the Custom Scripting tab.

Insert mediaDisplay.loadContent Command into a Custom Script

You can use a media display content file that you create through the Media Load Content option on the Components tab with a custom script simply by inserting the `mediaDisplay.loadContent` command into the script.

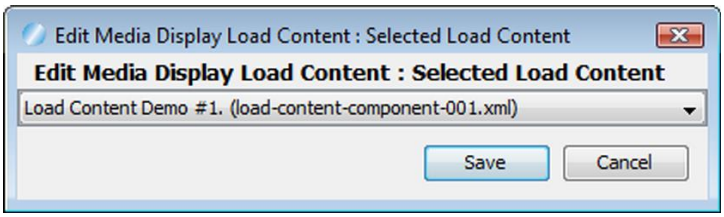
1. Go to: **Tester Toolkit** > **Custom Scripting**
2. Click **Add** to create a new script, or select an existing script and click **Edit**.
3. Scroll to **mediaDisplay.loadContent** in the **Available Commands** list.
4. Left-click and drag the command into the **Custom Script Commands** section.
5. Click the plus sign to expand the command information.
6. Double-click **Device ID**.
7. Click the **Variable DeviceId** drop-down arrow, and select the device(s) to which you want the command to go.

   Note: You can configure device variables through the Global Variables option on the Components tab.

8. Click **Save**.

   Note: When using other mediaDisplay.loadContent Commands (set ActiveContent and releaseContent) in a script, you must make sure they refer to the same content ID as was used in the loadContent component.

9. Double-click **Selected Load Content**.



7. Click the drop-down arrow, and select the media display file you want to load onto the EGM at the point you have inserted the `mediaDisplay.loadContent` command in the script.

   Note: If you do not see the media display content file you want to use in the drop-down list, go to the Media Load Content option and verify that it appears in the **Available Media Display Load Content Files** list.

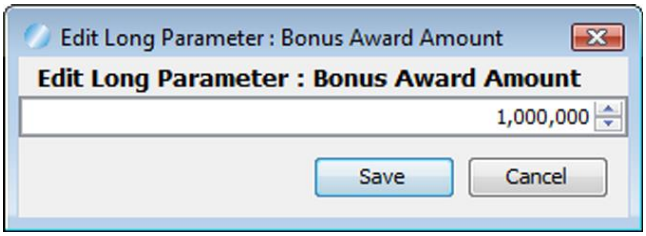
8. Click **Save**.
9. Click **Save** at the bottom of the **Edit a Custom Script** screen to save your changes and exit back to the Custom Scripting tab.

Insert bonus.setBonusAward Command into a Custom Script

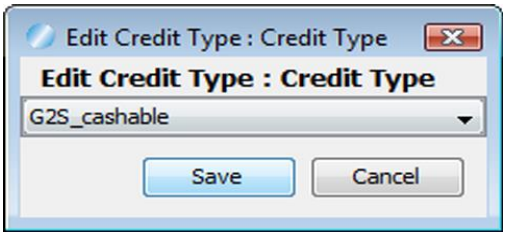
Inserting a `bonus.setBonusAward` command into a script is one way to easily put money on the EGM's credit meter during extended testing with a custom script. Once you insert the command into a script, you can configure its attributes as required.

1. Go to: **Tester Toolkit** > **Custom Scripting**
2. Click **Add** to create a new script, or select an existing script and click **Edit**.
3. Scroll to bonus.setBonusAward in the **Available Commands** list.
4. Left-click and drag the command into the **Custom Script Commands** section.
5. Click the plus sign to expand the command information.
6. Double-click **Device ID**.
7. Click the **Variable DeviceId** drop-down arrow, and select the bonus device(s) to which you want the command to be sent.

   Note: You can configure device variables through the Global Variables option on the Components tab.
8. Click **Save**.
9. Double-click **Bonus Award Amount**.

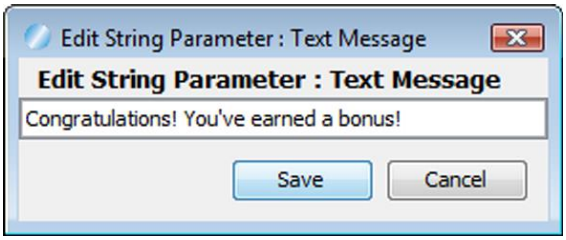


10. Select or type, in millicents, the amount of the bonus.
11. Click **Save**.
12. Double-click **Credit Type**.

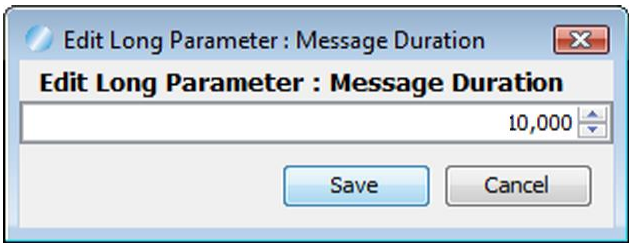


13. Click the **Credit Type** drop-down arrow, and select whether the bonus is cashable, non-cashable or promotional.

14. Click **Save**.
15. Double-click **Text Message**.



16. Type the message to be displayed at the EGM when the bonus is awarded.
17. Click **Save**.
18. Double-click **Message Duration**.

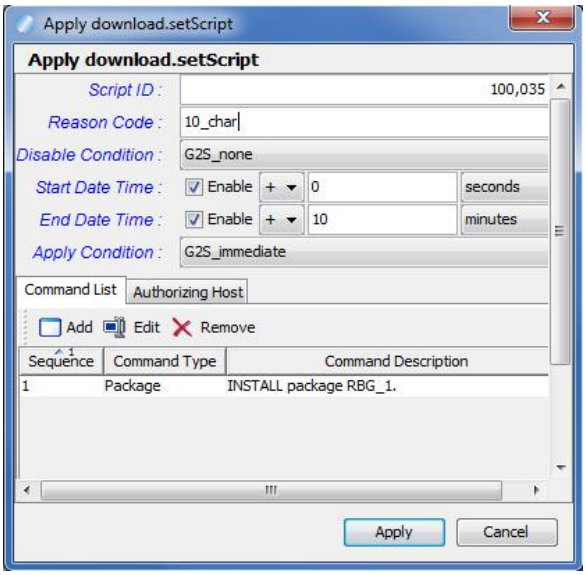


19. Select or type the time, in milliseconds, that the Text Message displays.
20. Click **Save**.
21. Click **Save** at the bottom of the **Edit a Custom Script** screen to save your changes and exit back to the Custom Scripting tab.

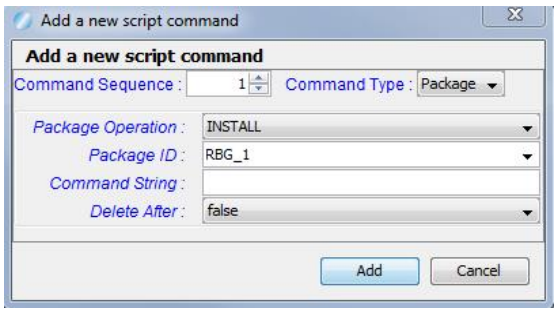
Insert **download.setScript Command into a Custom Script**

Inserting a `download.setScript`command into a script is an easy way to install a package for testing. Once you insert the command into a script, you can configure its attributes as required.

1. Go to: **Tester Toolkit** > **Custom Scripting** > **First GSA Script**, and click **Edit**.
2. In the **Available Commands** box, double-click **`download.setScript`**.
3. Complete the following **Apply download.setScript** fields, as needed:
   - **Script ID** - Unique value that is assigned automatically.
   - **Reason Code** - Enter up to 10 alphanumeric characters.
   - **Disable Condition** - Select when to disable the EGM.
     - **G2S_none** if disable condition does not apply.
     - **G2S_immediate** to disable the EGM immediately.
     - **G2S_idle** to disable the EGM when it is idle.
     - **G2S_zeroCredits** to disable the EGM when the credit meter reaches 0 (zero) credits.
   - **Start/End Date Times** - **Start Date Time** and **End Date Time** are optional fields. If times are enabled, those are the values executed each time this command is run.
   - **Apply Condition** - Select to indicate when to apply the script.
     - **G2S_immediate** to apply changes immediately after required authorizations are complete.
     - **G2S_disable** to apply changes after the EGM has disabled itself (`startDateTime`, `endDateTime`, and `disableConditions` apply).
     - **G2S_egmAction** to apply changes after an operator action at the EGM takes place, such as an audit key or menu option.
     - **G2S_cancel** to cancel changes after validation.
- Click the **Command List** tab.



- Click **Add**. Complete the following fields:
  - **Command Sequence** - Enter a number to put a run sequence in order .
  - **Command Type** - Here we selected **Package**.
  - **Package Operation** - Select **G2S_install** to install the specified package.
  - **Package ID** - Enter the unique identifier for the installable package. If you are using the RadBlue installable package, enter **RBG_[n]**.
  - **Command String** - (optional). Enter script command elements that you want to execute with the package. The EGM validates command strings prior to starting the script.
  - **Delete After** - If you select **false**, the package is deleted from the package list and internal storage once the command has completed.



4. Click **Add** to see your package ID/command description listed under the **Command List** tab.
5. Click **Apply**.
6. Click **Save**. At this point you can install scripts such as: setActiveDenom.

About setActiveDenoms

An owner host can send `setActiveDenoms`to set the list of denominations that can be played within a device.

When you send the `setActiveDenoms` command, you receive the device's current game combinations (`gameDenomList`), which are the game combinations available to the player.

You have the ability to send a `setActiveDenom` command using an RGS custom script by doing the following:

1. Download a new software package to the EGM by configuring package identifiers: size, type, and transfer location.
2. Install that package.
3. Indicate which denominations are to be active — use gamePlay.`setActiveDenoms`
   Note: You see this command displayed at **SendCommand >G2S_gamePlay> setActiveDenom**. This screen shows current EGM denomination ranges and is not supported by the `setActiveDenom`command in the Tester Toolkit tab.

Using setActiveDenoms

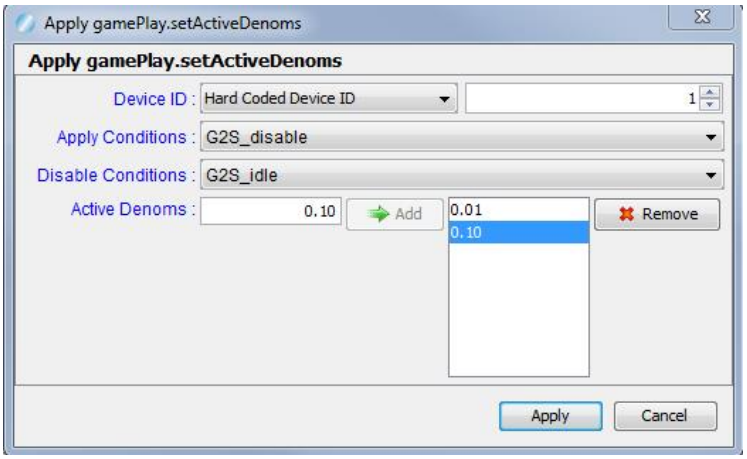
RGS uses this command to set the list of denominations that can be played within a device.

Follow these steps to use `setActiveDenoms`.

1. Go to **Tester Toolkit tab > First GSA Script**, and click **Edit**.
2. In the **Available Commands** box, double-click **`gameplay.setActiveDenoms`**.
3. Complete the following:
   - **Device ID**: Select one. In our example we select **Hard Coded Device ID**.
     - All Devices
     - First Device
     - Pick a Device
     - Hard Coded Device ID
   - **Apply Conditions**
     - **G2S_immediate** to apply changes immediately after required authorizations are complete.
     - **G2S_disable** to apply changes after the EGM has disabled itself (`startDateTime`, `endDateTime`, and `disableConditions` apply).
     - **G2S_egmAction** to apply changes after an operator action at the EGM takes place, such as an audit key or menu option.
     - **G2S_cancel** to cancel changes after validation.

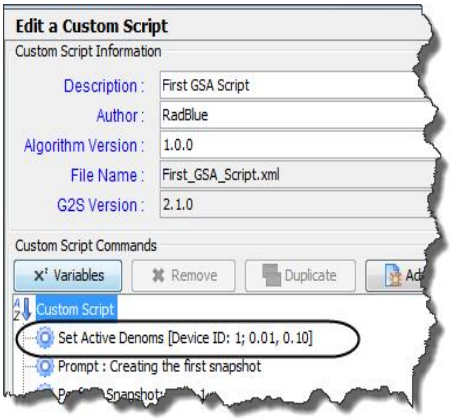
- **Disable Conditions**
  - **G2S_none** to disable condition does not apply.
  - **G2S_immediate** to disable the EGM immediately.
  - **G2S_idle** to disable the EGM when it is idle.
  - **G2S_zeroCredits** to disable the EGM when the credit meter reaches 0 (zero) credits.
- **Active Denoms** - Type one or more denominations to set as active, in millicents.

4. Click **Add** to move the denomination to the list. In our example, we typed **0.01,** clicked **Add**, typed **0.10**, and then clicked **Add**. Here you see both denominations added to the list.



5. Click **Apply**.

6. Look at the **Custom Script Commands** section of the screen. You see **Set Active Denoms** (showing your device ID and denomination selections).



7. Click **Save**.

About Script Verbs

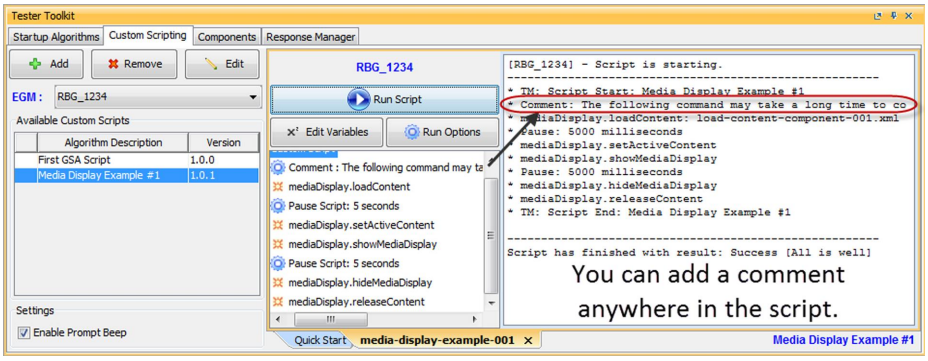
Script verbs perform script-specific functions (for example, pause script) within a custom script as well as complex commands. To use a verb, drag-and-drop it into the Custom Script Commands box, and configure as needed.

The following verbs are available with Custom Scripting:

- Comment
- Delete All Snapshots
- Enable All Host Disabled Devices
- Event Snapshots
- Notice
- Pause Script
- Perform Balanced Meters Analysis
- Perform Snapshot
- Perform Snapshot Compare
- Prompt
- Run Macro
- Run Script
- Send My Command
- Send My Raw Command
- Set Active Response Configuration
- Set Time To Live
- Wait For Commands
- Wait for Compares to Complete
- Wait For Events

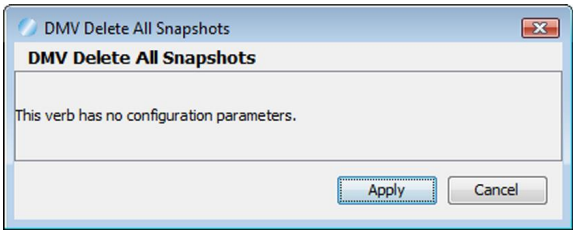
Comment

The Comment verb lets you add one or more comments to a custom script. You can add a comment anywhere within a script. The comment displays in the script viewer when you run the script.



Delete All Snapshots

The Delete All Snapshots verb deletes all prior data model snapshots that are stored in RGS. A list of snapshots is located in the data model viewer. This feature is useful when you are running a script that creates snapshots, and you want to start with an empty list. When you move this verb into the Custom Script Commands list, you see:



Click **Apply** to add the verb to the script.

Enable All Host Disabled Devices

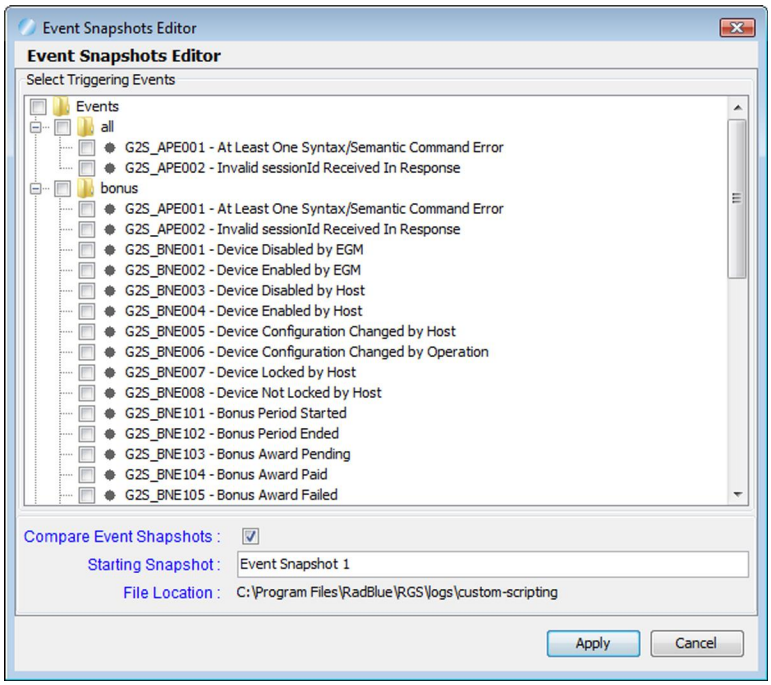
The Enable All Host Disabled Devices verb allows you to enable multiple devices. The host must be the owner of the device in order to enable/disable this change.

Edit this screen as needed. You can:

- Select **G2S_all** to enable all classes shown in the list, or select only one class from the list.
- Select **-1** to enable all devices.
  If you have two or more devices, you must select each device separately. For example, select device ID 2 and complete this box, click Apply. Repeat for device ID 4, and so on.
- Add a **comment**.
- Click **Apply**.

Event Snapshots

The Event Snapshots verb lets you define a list of events for which the RGS should capture a data model snapshot whenever any of these events are received by the tool, while the script is running. The snapshots are uniquely named using the form `eventCompare.eventCode.eventId`, so you can easily figure out which is which.



You can easily select and compare any of these event snapshots against a prior snapshot, using the Compare Data Models tab, located on the Databases layout.

You can also have the RGS automatically run a snapshot compare against the previous snapshot, while the script is running. These compare files are stored as XML files in the **../logs/custom-scripting** directory where you installed the product, in the form `eventCompare.eventCode.eventId.xml`. These XML compare files are very useful if you need to export the comparison data to an external system.

Notice

The Notice verb lets you create an informational message that is written to the Script Status window, at the point in the script you insert it.

Pause Script

The Pause Script verb pauses the script for the time, or within the time period, that you specify. Enter a Minimum Delay time and a Maximum Delay time in seconds. If the values are different, RGS pauses the script a random number of seconds between the minimum and maximum values. If both values are the same, RGS pauses the script for the number of seconds specified.

Perform Balanced Meters Analysis

The Perform Balance Meters Analysis verb runs a comparison of two EGM data model snapshots, providing a report on a variety of meter balancing tests. For a list of Balance Meters Analysis tests and a discussion of G2S meters, see "Appendix B" of the Gaming Standards Association's G2S Message Protocol document.

The Perform Balance Meters Analysis verb requires two different data model snapshots, which can be captured within the script. For example:

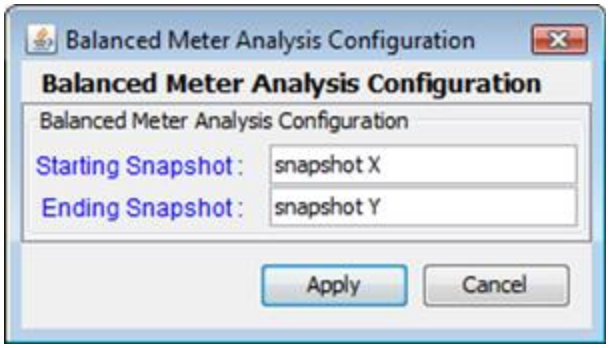
Perform Snapshot :snapshot X

Prompt: Please do a series of actions on the EGM in order to generate some G2S Events. (Button: Press When Done)

Perform Snapshot :snapshot Y

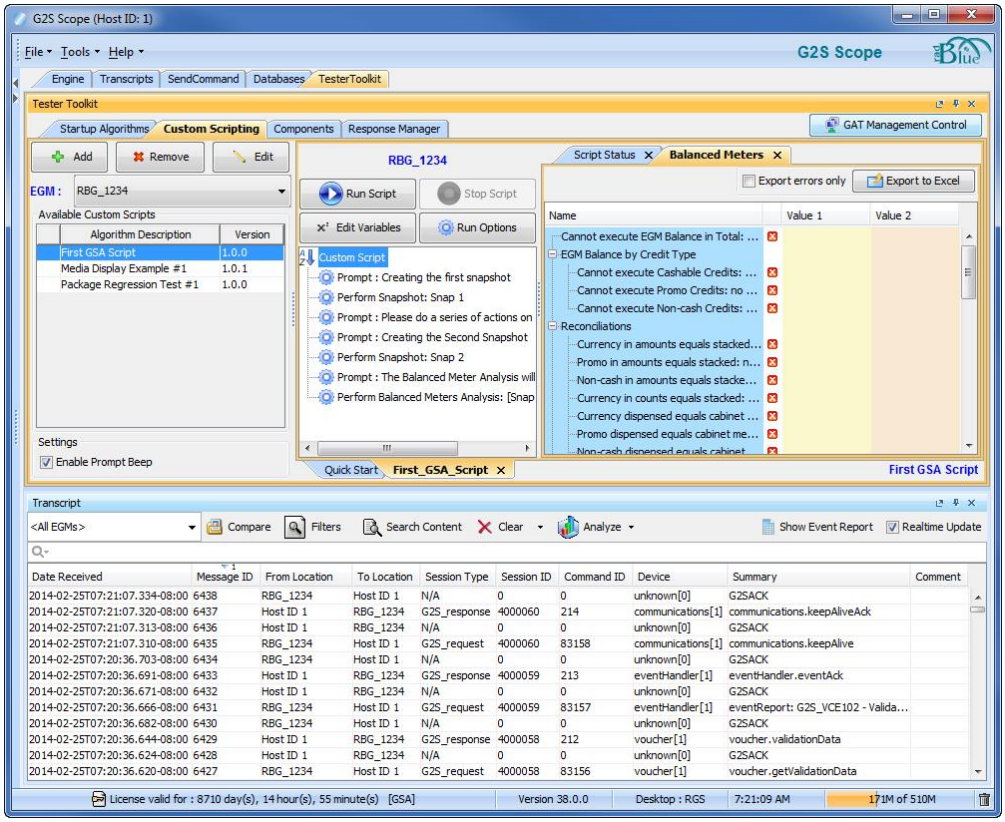
Perform Balanced Meters Analysis [snapshot X, snapshot Y]

When you move the Balanced Meter Analysis verb into the Custom Script Command section, you are prompted to enter the name of the starting snapshot and the name of the ending snapshot.



Note that you must enter a value for each field.

When the script has completed, a Balanced Meter tab displays the results of all tests that were performed against the changes in the meter data between the two snapshots.

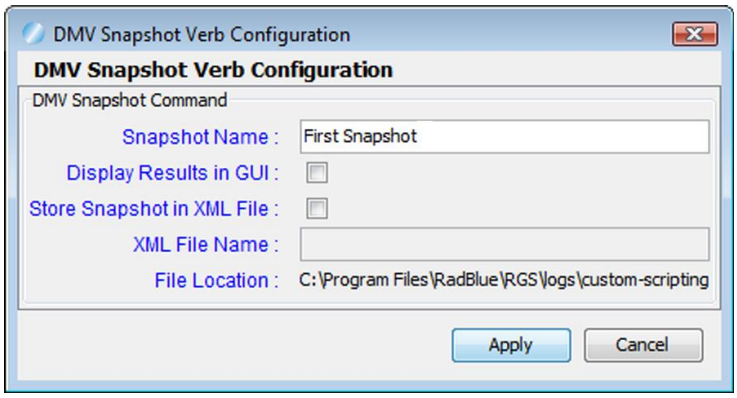


A status icon to the right of each field lets you locate imbalances quickly: a green circle with a checkmark means that the meters balanced; a red circle with an “x” means the meters didn’t balance.

Click **Export to Excel** to export the Balanced Meters Analysis Report. Select **Export errors only** to export only results with errors.

Perform Snapshot

The Perform Snapshot verb lets you take a snapshot of the EGM data model. You can use this verb multiple times to see changes as they occur over the course of testing.

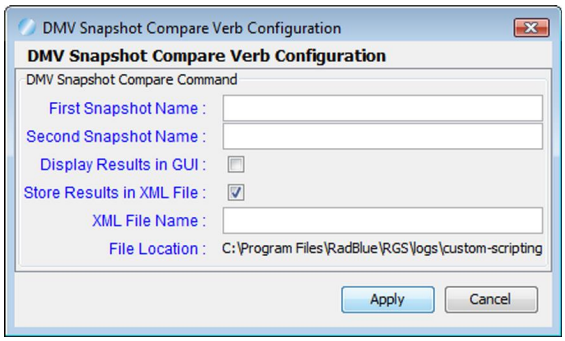


This verb requires a name for the snapshot. You can also indicate that you want the results displayed in the user interface (Display Results in GUI), which causes the script runner to open a new tab and display the standard data model viewer. If required, select to store the snapshot in an XML file, so you can easily import the file into another system.

Finally, you can manage snapshots from the Compare Data Models tab, located on the Databases layout.

Perform Snapshot Compare

The verb lets you automatically compare any two snapshots, displaying the results in a new script tab, while also optionally saving the results out to an XML file.



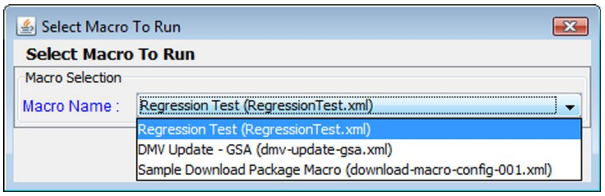
Prompt

The Prompt verb lets you create a custom prompt that displays to the user, in the Script Status window, at the point in the script you place it, at the point in the script you insert it. You can write a prompt description and define the button text. A timeout value lets you define how long, in seconds, the script waits for the button to be pressed before running the rest of the script commands.

Run Macro

The Run Macro verb lets you insert a wizard-driven script into your custom script.

When you move the Run Macro verb into the Custom Script Commands section, you are prompted to select a macro.



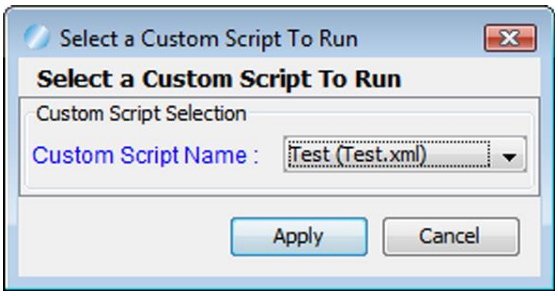
Click the drop-down arrow, and select from the available macros. A macro must be configured through the **Components** > **Macro** option to become available in this list. If you do not see the macro you are looking for in the drop-down list, go to the Macros option and verify that it appears in the Available Macros list.

The macro you select is run at the point in the script you insert it. Changes to macros must be made through the Macro option. Changes are applied immediately to macros that have been inserted into custom scripts.

Run Script

The Run Script verb lets you run one or more custom scripts within a custom script.

1. Move the **Run Script** verb to the place in the script you want the inserted script to execute.



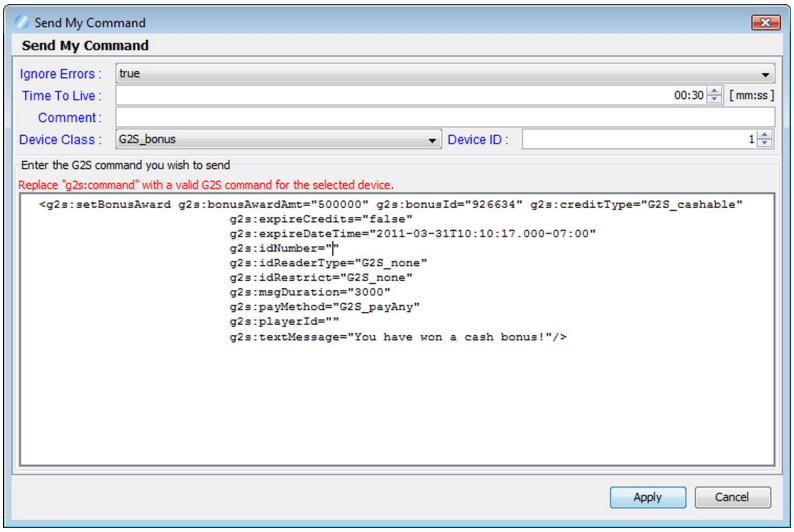
2. Click the **Custom Script Name** drop-down arrow, and select a script.
3. Click **Apply** to insert the script.

Send My Command

The Send My Command verb lets you send a custom command that RGS then wraps with the G2S class and G2S message elements before sending. The G2S information completed automatically by RGS includes command ID, session ID and date/time values. You can also add attributes and elements in custom namespaces, if required. This option is useful when you want to send a custom command to a device exposed through the descriptor list.

> **Note:** This script verb is available for Custom Scripting or Startup Algorithms. If using Send My Command with the Startup Algorithm, Send My Command runs at startup only if the *deviceID* is valid, or if it is listed in the `descriptorList` command. Otherwise, the Send My Command content is ignored by RGS, and the algorithm will continue.

1. Move the **Send My Command** verb to the place in the script that you want the command sent.



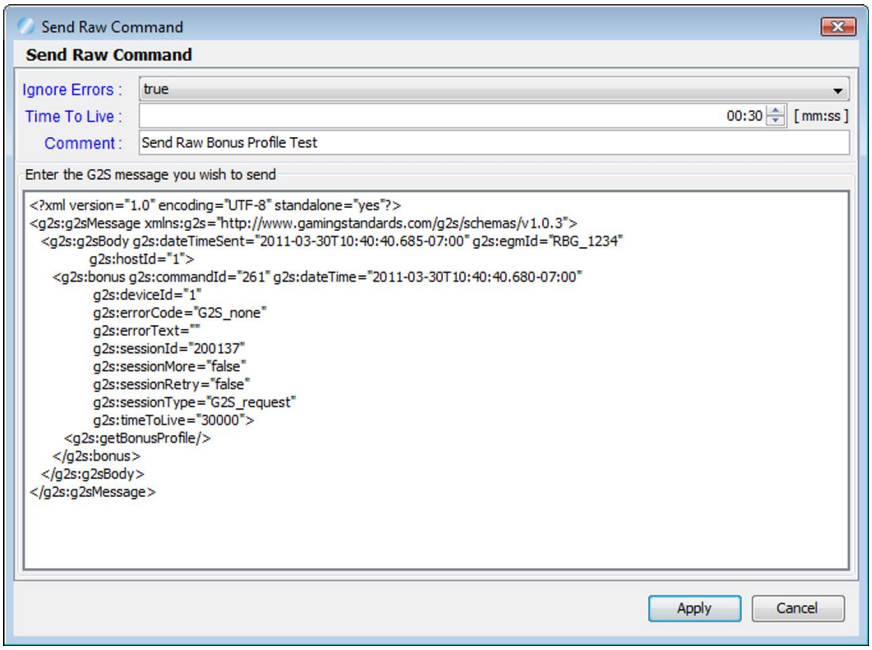
2. Set the verb options as needed.
   - **Ignore Errors** - Click the drop-down arrow, and choose whether RGS should validate the command information (false) or skip validation of the command (true).
   - **Time To Live** - This value is included as the timeToLive value in the command sent to the EGM (how long the command is valid for). The default value (typically 30 seconds) is used, unless you specify an alternate value.
   - **Comment** - Type identifying information for the command. This information appears in the Comment column of the Transcript.
   - **Device Class** - Click the drop-down arrow, select the appropriate class for this command.
   - **Device ID** - Type or select the device identifier of the device to which the command should be sent.

3. Click inside the text box.
4. Type, or cut-and-paste, the command content into the text box.
5. Click **Apply**.

Send My Raw Command

The Send Raw Command verb lets you send any content to an EGM over a G2S connection. RGS does no additional wrapping of the message, but rather, sends exactly what you have entered to the EGM.

1. Move the **Send My Raw Command** verb to the place in the script that you want the command sent.



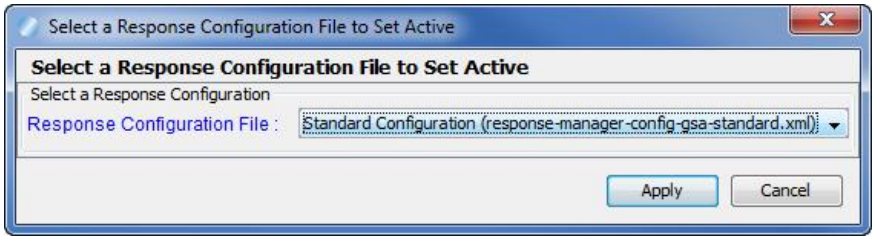
2. Click inside the text box.
3. Type, or cut-and-paste, the message content into the text box.
4. Click **Apply**.

Set Active Response Configuration

The Set Active Response Configuration script verb lets you set an active custom response from a custom script by selecting the response configuration you want to add.

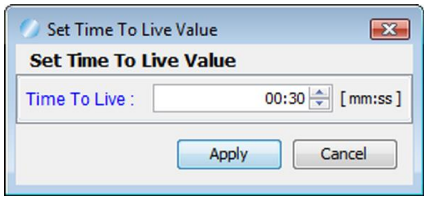
Once you add a response configuration file through the Response Manager, it becomes available in the Set Active Response Configuration drop-down.

To add a response configuration to a custom script, click the **Response Configuration File** drop-down and select the response configuration you want to insert. Then, click **Apply**.



Set Time to Live

At times, it may be necessary to change the *timeToLive* attribute for messages sent within the script. This verb changes the default *timeToLive* value used throughout the tool when constructing G2S messages.

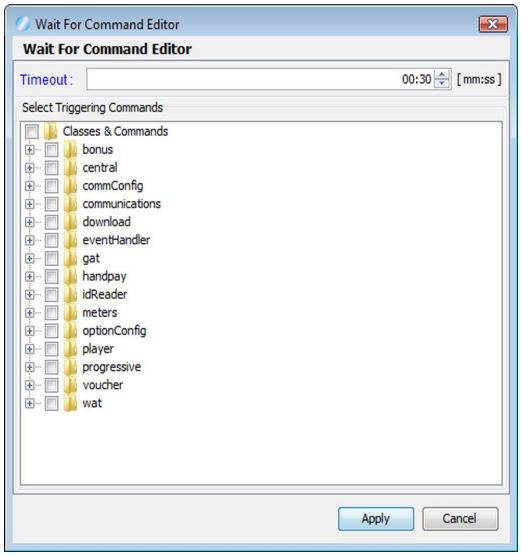


Using this verb is essentially the same as changing the configuration option, and affects the *timeToLive* value used in all subsequent commands sent by the RGS.

Wait for Commands

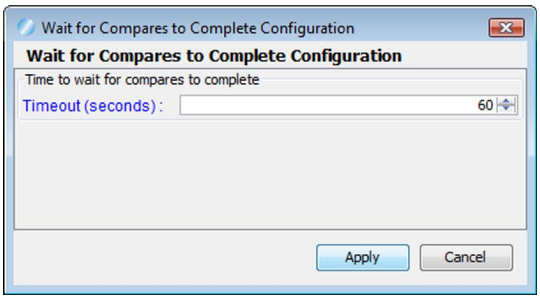
The Wait for Commands verb causes the script to pause until any of the selected commands are received by RGS. This is especially useful if you are invoking extended sequences on the EGM (such as installing a new package), and you want to have the script wait until the RGS restart sequence has completed.

> **Note:** This command is not intended to allow you to write your own custom responses to received messages. Custom response functionality is available through the Response Manager.



Wait for Compares to Complete

If you are using the Event Snapshots verb, and it is causing a lot of event snapshots and compares to run, you may want to have the script wait until all of the compare operations are complete, before moving on to the next phase of the script. This verb does that, and a timeout is provided so you can specify the maximum time for the script to wait for all of the compares to complete.



Wait for Events

The Wait for Events verb causes the script to pause until any of the selected events are received by the RGS.

About the Response Manager

The Response Manager lets you customize how RGS responds to G2S commands sent from the EGM. First, you create a response configuration, which consists of one or more custom responses. Next, you add custom responses to the response configuration. A custom response is a predefined response for a specific command.

For each custom response, you must define the following:

- the command that triggers the custom response
- the response action
- the response pattern

The response action is the action taken when RGS receives the specified command:

- send no response (no action)
- send a custom command
- send a predefined application error
- send a custom application error

The response pattern defines which occurrence of a command the custom response should be applied to and how many times total the custom response should be applied.

For example, a 2/5 pattern would look like this:

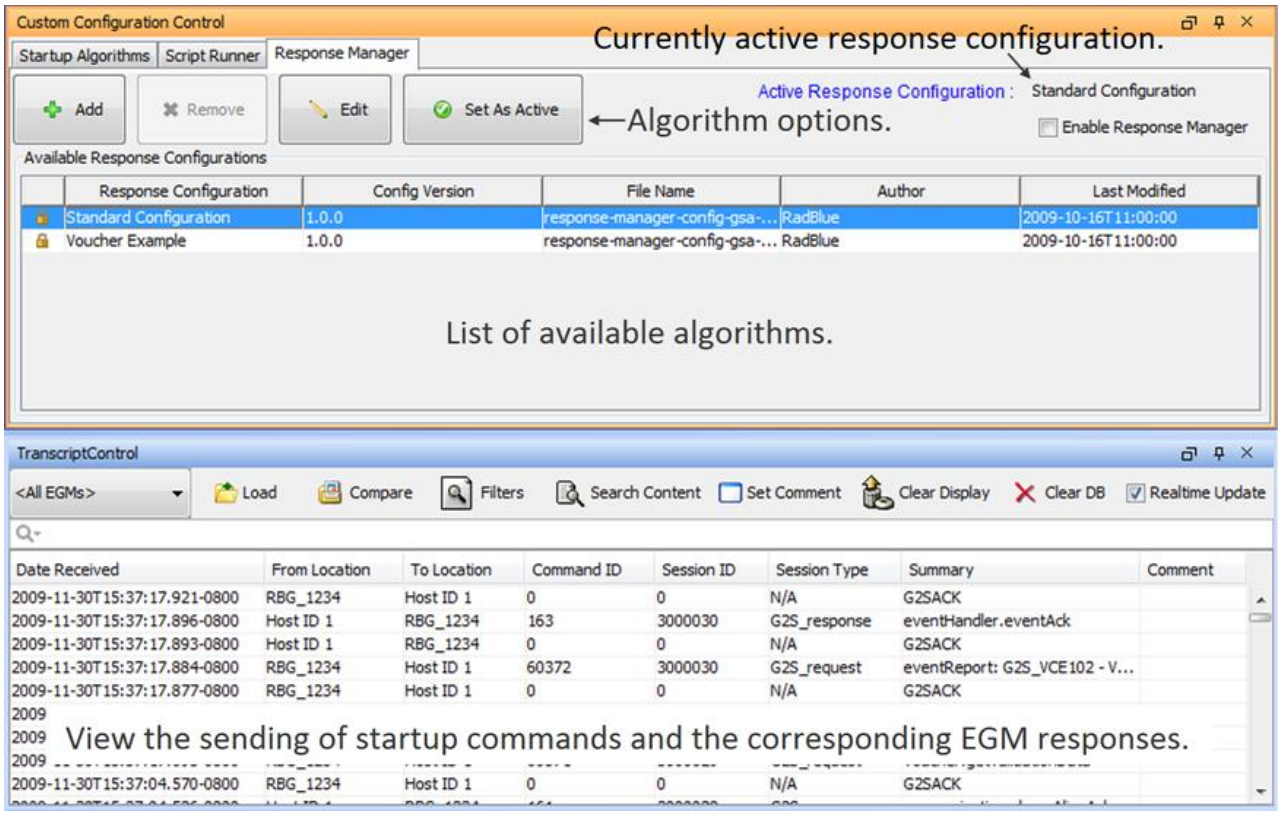
1st - normal response*	10th - **custom response**
2nd - custom response	11th - **custom response**
3rd - custom response	12th - **custom response**
4th - custom response	*13th - normal response*
5th - custom response	**14th - custom response**
6th - custom response	**15th - custom response**
7th - normal response	**16th - custom response**
8th - custom response	**17th - custom response**
9th - custom response	**18th - custom response**

**Instance of command received by RGS.*

The 2/5 pattern indicates that the response will be applied the second time the command is received by RGS as well as to the next four instances of that command. You can choose to repeat the pattern for as long as the response configuration is enabled.

Review the Response Manager Layout

The Response Manager tab is located on the Tester Toolkit layout. The Response Manager tab contains the Response Manager and the Transcript object.



Set Active Response Configuration

Before you can use custom responses, you must set the active response configuration.

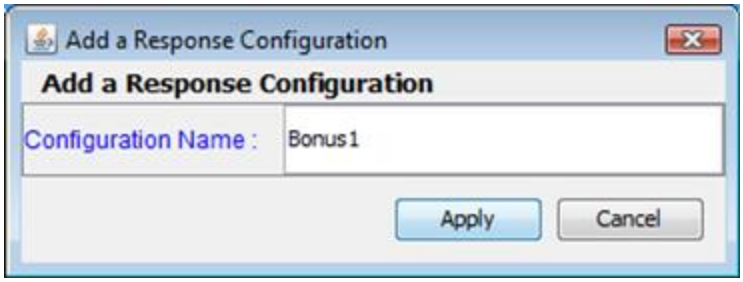
1. Go to: Custom Configuration > **Response Manager**.
2. From the **Available Response Configurations** list, click to select the response configuration you want to set as active.
3. Click **Set As Active**.
4. Verify that the **Active Response Configuration** field displays the description of the response configuration you selected in step 1.
5. Verify that **Enable Response Manager** is selected if you want RGS to use your selected response configuration.

Add or Edit a Response Configuration

Use this procedure to add or edit a response configuration. Each response configuration consists of one or more custom responses.

See About the Response Manager.

1. Go to: **Custom Configuration** > **Response Manager**
2. To create a new response configuration:
   a. Click **Add**.

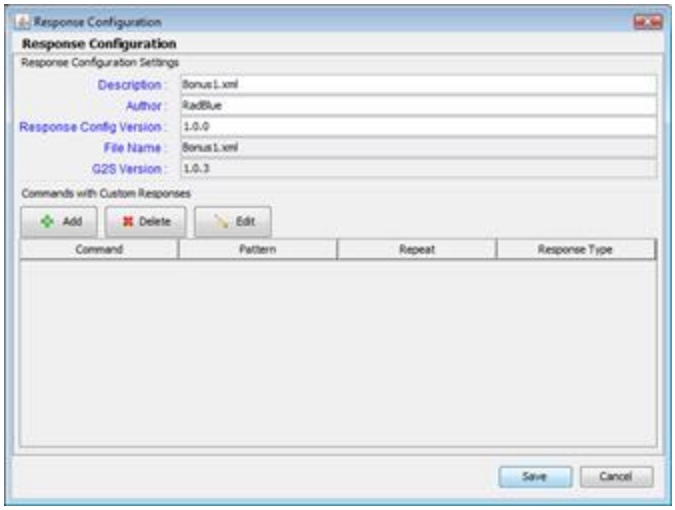


   b. Type a name for the new response configuration (spaces are not allowed). Note that the file name must be unique.
   c. Click **Apply**.

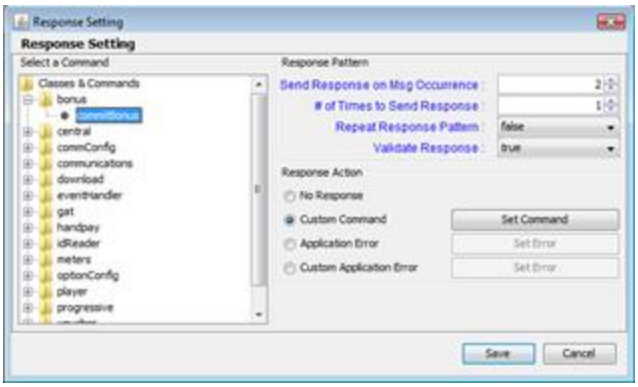
or

To open an existing response configuration to edit:

   a. Click to highlight the response configuration you want to edit.
   b. Click **Edit**.



3. Modify information for the response configuration. Note that RGS automatically inserts default information.
   - **Description** - Type a description for the response configuration.
   - **Author** - Type an identifier for the response configuration's author.
   - **Response Config Version** - Type a version for number for the response configuration. This field can contain any information required.
   - **File Name** - File name of the response configuration (entered in step 2b of creating a new response configuration). This field is *read-only*.
   - **G2S Version** - Indicates the selected version of the Game-to-System (G2S) protocol. The G2S version is defined in the Schema Option field under Configure > Engine Options. This field is read-only.
4. Click **Add** to add a command to the response configuration.

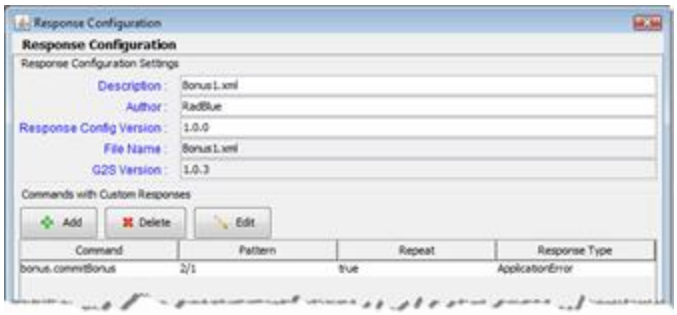


5. Select a command for the response by expanding the class and clicking to select the command. Note that only commands that are received by RGS are available.
6. Enter the **Response Pattern** information for the selected command.
   - **Send Response on Msg Occurrence** - Type or select when the response for the specified command is applied. For example, **2** indicates that the response will be applied to the second command received.
   - **# of Times to Send Response** - Type or select the number of times, including the first occurrence, to send the response. For example, **1** indicates that the response will be applied only once (to the first occurrence).
   - **Repeat Response Pattern** - Click the drop-down arrow, and select **true** to repeat the pattern created by the **Send Response on Msg Occurrence** and **# of Times to Send Response** values. Select **false** if you do not want the pattern to repeat.
   - **Validate Response** - Click the drop-down arrow, and select **true** if you want RGS to validate the outbound responses for the selected command. Select **false** if you do not want RGS to validate the outbound response.

7. Select a **Response Action** for the selected command.
   - **No Response** - Select if you do not want to send a response to the specified command.
   - **Custom Command** - Click **Set Command**, enter custom command information, and click **OK**.
   - **Application Error** - Click **Set Error**, click the drop-down arrow and select the application error you want to send, and click **Select**.
   - **Custom Application Error** - Click **Set Error**, type the error code and error text you want to send, and click **Select**.

   Note: The entered error code must be in valid G2S format - 3-character manufacturer ID, an underscore ("_") and the 6-character error code. For example, **xxx_yyyyyy**.

8. Click **Save**.

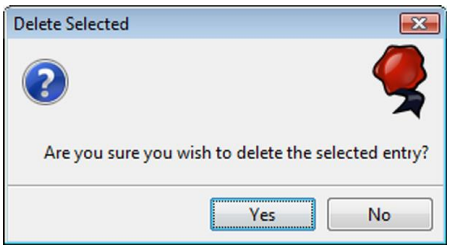


9. Once you have added or edited the commands in the response configuration as needed, click **Save**. If the response configuration is new, it is added to the **Available Response Configurations** list.

Delete a Response

You can delete any response configuration that you create. RadBlue default and sample response configurations (for example, Voucher Example) are read-only and cannot be removed.

1. From the **Available Response Configurations** list, click to select the response configurations you want to delete.
2. Click **Remove**.



3. Click **Yes** to delete the selected response configuration, or click **No** to return to the main screen without making any changes.

About GAT Software Verification

The **GAT Management Control** feature, on the Tester Toolkit layout, can be used as a sample GAT software verification tool. The GAT Management Control consists of two parts: the **GAT Component Digest** and the **GAT Component Verification**.

Create a GAT Component Digest

From this tab, you can create a signature digest file for the software on a known EGM that can later be used to verify the software signatures on *another* EGM to make sure that EGM is running the same software (or at least is able to generate the same signatures).

To create a new digest file:

1. Select the **Tester Toolkit** layout, and click **GAT Management Control**.
2. Select the **GAT Component Digest** tab.
3. Select **New**, and enter a file name and description for the new digest file.
4. Click **Create New Component Digest File**. A blank digest file is created.
5. Select a **Connected EGM** that will be used as the reference instance
6. Select **Verify Components**, which launches a new object that contains a listing of all components in the selected EGM, as were reported by the EGM during the start-up algorithm.
7. Select arow, and click **GAT doVerification** to have RGS instruct the EGM to verify the selected component using the provided values.

   or

   Select **Automate GAT doVerification** to have the RGS automatically process all unverified rows in the table (all algorithms for all components in the EGM).

   Where Seeds, Salts or Offsets are supported for an algorithm, RGS automatically provides a value (displayed as columns in the table). As each verification result is returned, the **Verify State** moves to **G2S_complete**, and this component can then be added to the digest.

   As each component–algorithm combination is verified and a result is returned by the EGM, the row changes to light purple shading, indicating that this value can be added to the digest file.
8. Once all selected rows have been verified, select **Add to Digest** to move this set of component verification results into the digest file for later use. This set now forms the digest of the software signatures of the known-good software on the control EGM.

Use GAT to Verify Components

The GAT Component Verification tab is used to compare the verification results for selected components in a new machine against the results from known-good software on the control EGM.

To verify the components against a GAT Component Digest file:

1. Select the **Component Digest File** that contains the known-good software results. If a newly created file is not in the drop-down list, press the **Refresh** to reload the file list.
2. Select an EGM that is connected to the RGS. When the EGM is selected, the **GAT Components** table is populated with a listing of each of the software components in the EGM. The RGS then randomly selects an algorithm for each component from the set of combinations that are common between the EGM under test and the GAT Component Digest File.

   Note: To edit the verification record of a component, double-click the row containing the component you need to change. Then, select an alternate **Algorithm Type**, or exclude this component from the verification.
3. Once you are satisfied with your selections, press **Start Verification** to begin the verification process of the selected components.

   For each verification, the stored Salt, Seed and /or Offset for the selected algorithm of each component is sent to the EGM in the `gat.doVerification` command. The EGM is then expected to return the same result that is stored in the selected **Component Digest File**.

   As each result is returned by the EGM, the **Component Result** column is updated with the result of the verification. The results can then be Exported to an Excel report, if needed.

V