RLT 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.

Contents

About the RLT Installation

The RLT is only supported on a Windows 64-bit operating environment.

Pre-Installation Requirements

1. The computer you install RLT on must have a network connection. If it does not, you will not be able to send multicast commands successfully.
2. You must have the RLT license file on your computer prior to installing RLT. If you are using a special version of RLT, you must have a license for that version. If you have not received an RLT license file, contact RadBlue Support.

Computer Requirements

The minimum requirements for computers running the load tester are:

- Operating System: 64-bit version of Windows Vista, 7, or Windows Server 2003
- Memory: 8 GB
- Disk space: 250 MB
- Processor: quad or dual quad

Since the memory footprint of each EGM depends on the size of its data model, the number of EGMs that run in this configuration will vary. Contact RadBlue to discuss the various RLT configurations available.

Install RLT

Follow these steps to install RLT on a Windows 64-bit operating system.

1. Double-click **RLT_x.x.x**.
2. Click **Next**.
3. Select **I accept the agreement**, and click **Next**.
4. Type the installation location of the RLT application, or click **Browse** and navigate to the location.
5. Click **Next**.

   Note: If you have a previous version of the tool installed in the same directory, you are prompted to remove it before installing the new version. Click **Yes** to uninstall the previous version before continuing with the new installation, or click **No** to install the new version in a different directory.

6. Navigate to the location of the **Load Tester license file**, and click **Next**.

   Note: For version 34 and higher, if you install a version of RLT over an existing version, you can choose to use the existing license. If you do not want to use the existing license, you can browse to a new license. Note that this option is only available when you install RLT over a previous installation. All components of the previous installation are removed by the installer except the license file and any backup files.

7. Select **Start** options.
   - Select the Start menu folder for RLT.
   - Select whether you would like RLT shortcuts created.
   - Select whether you would like RLT shortcuts created for all user accounts on the computer.
8. Click **Finish**.

Uninstall RLT

You can uninstall RLT through the **Uninstall** option (**Start** > **All Programs** > **RadBlue Load Tester**) or by running the **uninstall.exe** file in the RLT installation directory.

When RLT is uninstalled, a backup folder is created in the RLT directory that saves the installation's security and configuration parameters. When a subsequent RGS version is installed, the installer uses the backed up data to populate security and configuration settings, so you do not need to re-key the information into the new installation. Additional files are also saved, so you can retain EGM and configuration information between versions.

> **Note:** Backup files are **not** available for versions using student licenses, which always use default values for upgrades.

Backup files are located in the RLT installation directory. The following information is saved when RLT is uninstalled:

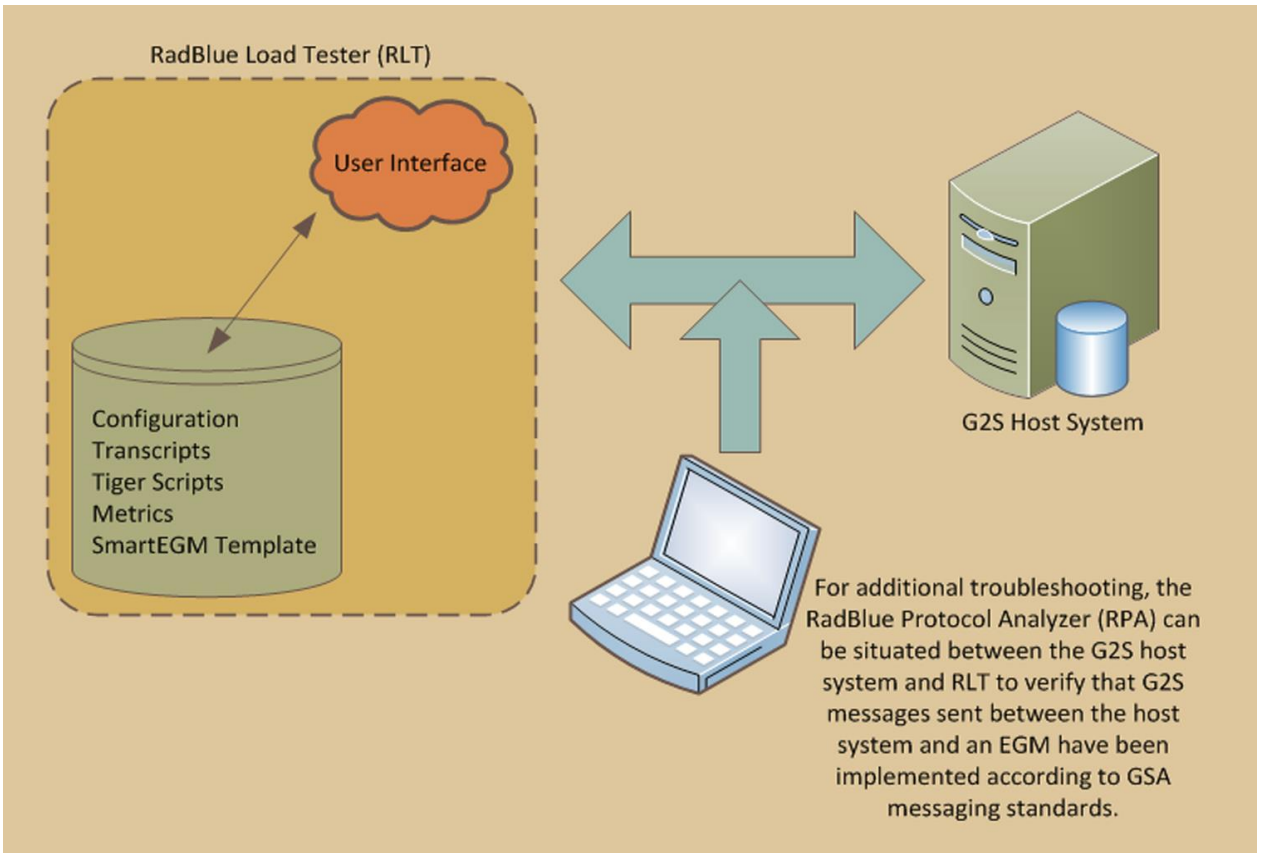
- all Java Keystore Files (JKS)
- database containing EGMs, Tiger Scripts and configuration files
- configuration options
- RLT license

About RLT

The RadBlue Load Tester (RLT) is a multi-threaded G2S load-test application that lets you evaluate the performance of a system or application, as well as the impact of a new feature on the performance of a G2S host system.

With RLT, you can:

- configure a G2S host to emulate up to 200 EGMs, all asynchronously sending messages to the system under test.
- choose whether to have all message traffic flow through a single socket, or add multiple IP addresses to create a pool of IP addresses that are equally shared among all EGMs.
- run multiple engines (on one or more servers, depending on your testing needs), which contain multiple simulated EGMs. All of these simulated EGMs communicate with your G2S host system.



RLT message flow

Each EGM contains a complete implementation of the SmartEGM engine, so it has its own data model and can run its own Tiger scripts - exactly the same scripts that are used in the RadBlue System Tester (RST). As with RST, all 22 G2S classes are supported and each EGM has its own Tiger script for automated testing.

> **Note:** Since the Load Tester shares the SmartEGM core with the System Tester, any changes made in our core G2S implementation are inherited by the Load Tester.

Scalable

Each RLT engine is scalable from one to 200 EGMs, with each EGM having a custom SmartEGM configuration and Tiger script. If you need more, add more instances.

Additional Resources

- RLT Release Notes

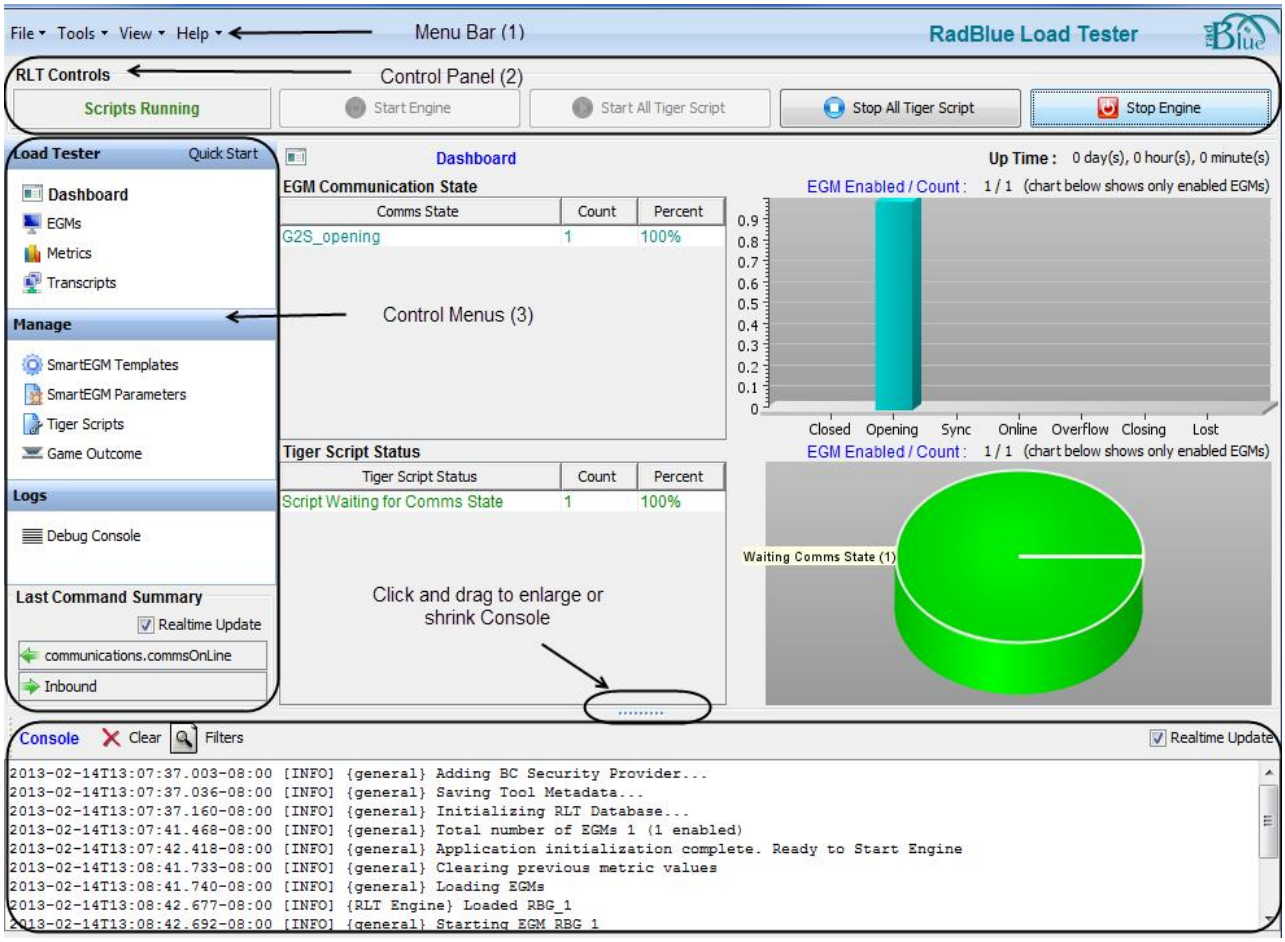
Supported GSA Versions

The following Gaming Standards Association (GSA) protocol versions are supported by RLT:

Protocol	Versions		
G2S	1.1.0	2.1.0	

Review the RLT User Interface

Let's look at the interface.



Menu Bar (1)

Option	Description
Export Metrics	Export metrics collected by RLT(and displayed on the Dashboard) to an XML file, from the beginning of a run until you select this option.
Export Debug	Select to create a ZIP file of troubleshooting information that can be sent to RadBlue support or used with the RadBlue Analysis Suite (RAS). Browse to the location where you want to save the ZIP file, and click **Save**.
Exit	Select to close the product.

Tools

Option	Short cut	Description
Configure	F2	Select to see configuration options.
GSA Message Validator	F4	Allows you to paste in a sample XML document and see if the message is valid against the selected schema.

View

Option	Short cut	Description
Toggle Display Mode	F11	Click to only show the Console message display - closing all functions except the Console. Displaying just the Console improves message processing.

Help

Option	Description
RLT Help	Select to Launch RLT Help system.
Contact Us	Select to open the contact page of the RadBlue web site.
About RLT	Select to see the copyright, licensing, and version information.

Control Panel (2)

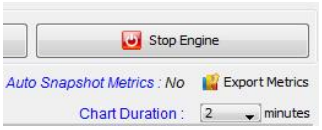
From RLT Control you can:

- Start and stop all EGMs and automated (Tiger) scripts associated with the EGMs.
- See metrics and diagnostics for all EGMs using the **Load Tester Menu** and Dashboards.
- See all messages sent and received by RLT through the Message Transcript.
- See all warnings, errors and informational messages generated by RLT using the **Console** window.
- Import a custom EGM data model (SmartEGM configuration template, and parameters, Tiger scripts) and assign it to EGMs, using the **Manage Menu**.
- Import automated test scripts and assign to EGMs.configure game outcome.
- Configure game outcome.

Menus (3)

Load Tester Menu - From the Load Tester menu you can (Dashboards),

- **Dashboard** - Click to see EGM communications and Tiger Script status information. You can see EGM details displayed graphically, as a percent and as a count.
- **EGMs** - Click to manage the EGMs. You can add, edit, or delete an EGM from a test and see EGM details -description, communication state, the number of EGMs in each communications state (for example, 12 EGMs in a G2S_lost state).
- **Metrics** - Click to see all metrics information.
  - **Auto Snapshot Metrics** interval time. This feature lets you automatically save RLT metrics and export them to a file. Use the Snapshot Interval to specify the time between snapshots. Click **Export Metrics** to save the file.



  - **Metric Delta** displays the number of events generated, messages sent and received, and Tiger verbs executed in a real-time graphical interface.
  - **Metric Totals & Averages** provides totals for the events in the queue, total events generated, various errors, messages sent and received, Tiger scripts executed with errors, Tiger scripts executed successfully and all Tiger scripts executed.
- **Transcripts** - See message information for a selected EGM.

Manage Menu - From the Manage menu you can manage SmartEGM templates and parameters, Tiger scripts and game outcome information.

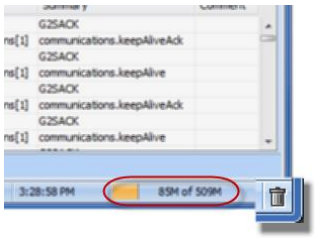
- **SmartEGM Templates** - See current EGMs listed, and modify file and host information.
- **SmartEGM Parameters** - Import SmartEGM templates; and load, edit, or delete all parameters.
- **Tiger Scripts** - Import Tiger scripts; and add or delete the scripts.
- **Game Outcome** - If you are using Game Outcome extension, you can import outcome themes and use them in combination with a modified SmartEGM template, to exercise Game Outcome on a host system.

Logs - See message error information.

- **Debug Console** - Shows error message information.

About the Garbage Collector

The Garbage Collector lets you reclaim memory that is no longer in needed in order to improve tool performance. To use the Garbage Collector, click the garbage can ( ) icon in the lower right corner of the user interface.



Get Started Using RLT

The following information assumes that you have already added EGMs to RLT, using the default sample configuration.

1. Configure the EGM connection to the host(s).

   If you are not using your own custom EGM configuration file, you can quickly modify the target host(s) used by the default configuration file. Note that the default configuration file is associated with all of your newly created EGMs.

   a. Click the **SmartEGM Templates** option on the **Manage** menu.

   b. Click to select the **Sample – RadBlue SmartEGM** template from the **Manage SmartEGM Templates** list. Directly below the list, the host and device information for the configuration template display.

   c. To edit a host, select the host and click **Edit Host**. Change the host identifier, description and URL for the host system as needed.

   d. Click **Save**.

   e. Repeat this process for as many hosts as required.

2. Verify that RLT is using the correct "from" address.

   Go to: **Tools** > **Configure** > **Engine Options** > **Transport** > **Bind To Address**. If not, click the drop-down arrow and select the correct address.

3. Return to the **Dashboard**.

4. To start all of your EGMs, click **Start Engine** under RLT Controls.

   Note: Only EGMs with the **EGM Enabled** field set to **true** will start. If an EGM's **EGM Enabled** field is set to **false**, that EGM will remain stopped when you click **Start Engine**. You can change the enabled status of an EGM through the **EGM Details** screen. See "Edit an EGM" on page 25.

   Once you have started all EGMs, you can stop and restart each EGM individually by right-clicking the EGM entry on the **EGMs** > **Manage EGMs** screen.

   Use the **Console** at the bottom of the screen to view activity and errors.

5. Look at the **EGM State Details** on the Dashboard, and confirm that all EGMs move to the **G2S_online** state.

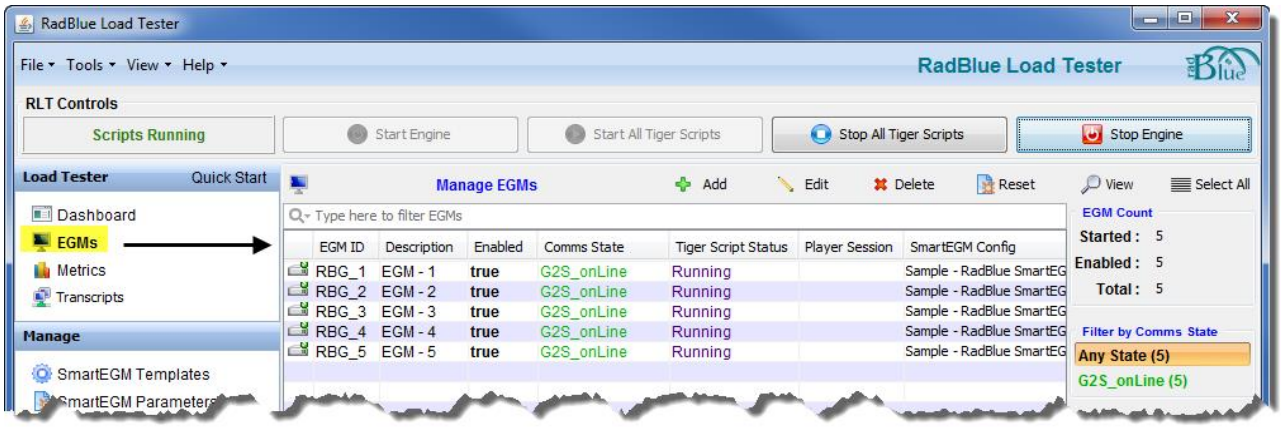
6. Once all of your EGMs are online, start automated testing by clicking **Start Tiger Script** under RLT Controls.
   - The **Last Command Summary** lets you see that commands are being sent between RLT and the host system(s). Clear the **realtime update** checkbox as needed to stop the flow of inbound and outbound messages in the display to better view the messages that are being sent between RLT and the host system.
   - **Metric Deltas**, accessed by clicking the **Metric Deltas** bar, displays specific message types (sent, received and Tiger) and the number of events queued in a real-time graph. Click the Duration drop-down to change the amount of time displayed by the graph.
   - **Metric Totals**, accessed by clicking the **Metric Totals & Averages** bar, displays values for various metrics, including event queue size and count, several errors, total messages received and total messages sent. In addition, it displays averages for messages sent for each EGM and `g2sAck` response times.
   - **Tiger Script Status**, accessed by clicking the **Tiger Script Status** bar, displays the current status for the Tiger Scripts associated with all EGMs (whether currently communicating with the host system or not).
   - **EGM Communication State**, accessed by clicking the **Communication State** bar, displays the communication state for all EGMs. State options are: closed, opening, sync, online, closing and lost.

About EGMs

From the **EGMs** option on the Load Tester menu, you can add up to 200 EGMs. Each EGM can have a separate automated (Tiger) script and data model (SmartEGM template), or all EGMs can use the same information. After the initial start of all EGMs, you can then stop and start each EGM individually. In addition, you have the option of closing the RLT interface while continuing to run EGMs.

You can do the following using the control panel:

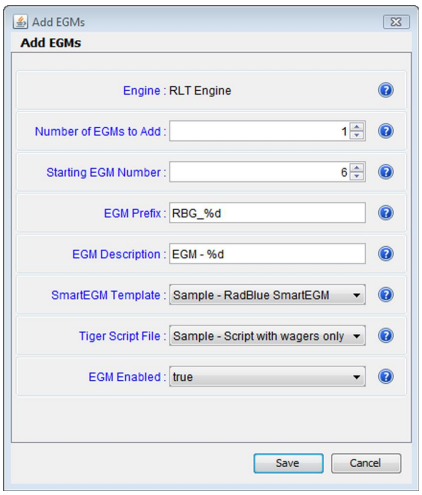
- **Add** - Add a new EGM to RLT.
- **Edit** - Change the EGM description, SmartEGM Config file, or Tiger Script.
- **Delete** - Remove an EGM from the RLT.
- **Reset** - Reset the EGM(s) data model to the original SmartEGM template configuration.
- **View** - See the changes and the last date that the EGM was modified.
- **Select All** - Change all EGMs configuration at the same time.



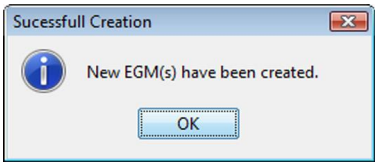
Add EGMs to RLT

You can add new EGMs to RLT through the EGMs option.

1. From the **Load Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. Click **Add**.



3. Type or select the **Number of EGMs to Add**.
4. Type or select the starting number for the new EGM(s).
5. Type a prefix for the EGM(s).
6. Type a description for the new EGM(s).
7. Click the **SmartEGM Template** drop-down arrow, and select the data model associated with the new EGM(s).
8. Click the **Tiger Script File** drop-down arrow, and select the Tiger script associated with the new EGM(s).
9. Click the **EGM Enabled** drop-down arrow, and select whether the new EGM(s) will be enabled (**true**) or disabled (**false**).
10. Click **Save** to add the new EGM(s).



11. Click **OK** to confirm the creation of the new EGM(s).

    The new EGM(s) are added to the list of EGMs.

Edit an EGM

You can edit EGMs through the EGMs option.

1. From the **Load Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. Click to select the EGM you want to edit.

   or

   CTRL+click to select multiple EGMs to edit.

   or

   Click **Select All** if you want to edit all EGMs.
3. Click **Edit**, or right-click and select **Edit**.
4. Modify the EGM settings as required.
   - **EGM Description** - Type a new description for the EGM. If you are editing multiple EGMs, this field is disabled.
   - **EGM ID** - Type a unique EGM identifier. This field is *read-only*.
   - **SmartEGM Config File** - Click the drop-down arrow, and select a new SmartEGM template.
   - **Tiger Script** - Click the drop-down arrow, and select a new Tiger script.
   - **EGM Enabled** - Click the drop-down arrow, and select **true** to enable the EGM(s) or **false** to disable the EGM(s).
   - **Reset EGM Data Model** - Select to reset the EGM Data Model to the original SmartEGM template configuration.
   - **EGM Data Model** - Click **Show** to view the current EGM data model. If you are editing multiple EGMs, this option is not displayed.
5. Click **Save**.

View EGM Details

You can view an EGM's data model, status, associated Tiger script and SmartEGM template, communication state and the last date that the EGM was modified through the EGMs option. If you select multiple EGMs to view, a list of those EGMs appears on left of the viewer so you can click through them without having to exit the viewer.

1. From the **Loads Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. Click to select the EGM you want to edit.

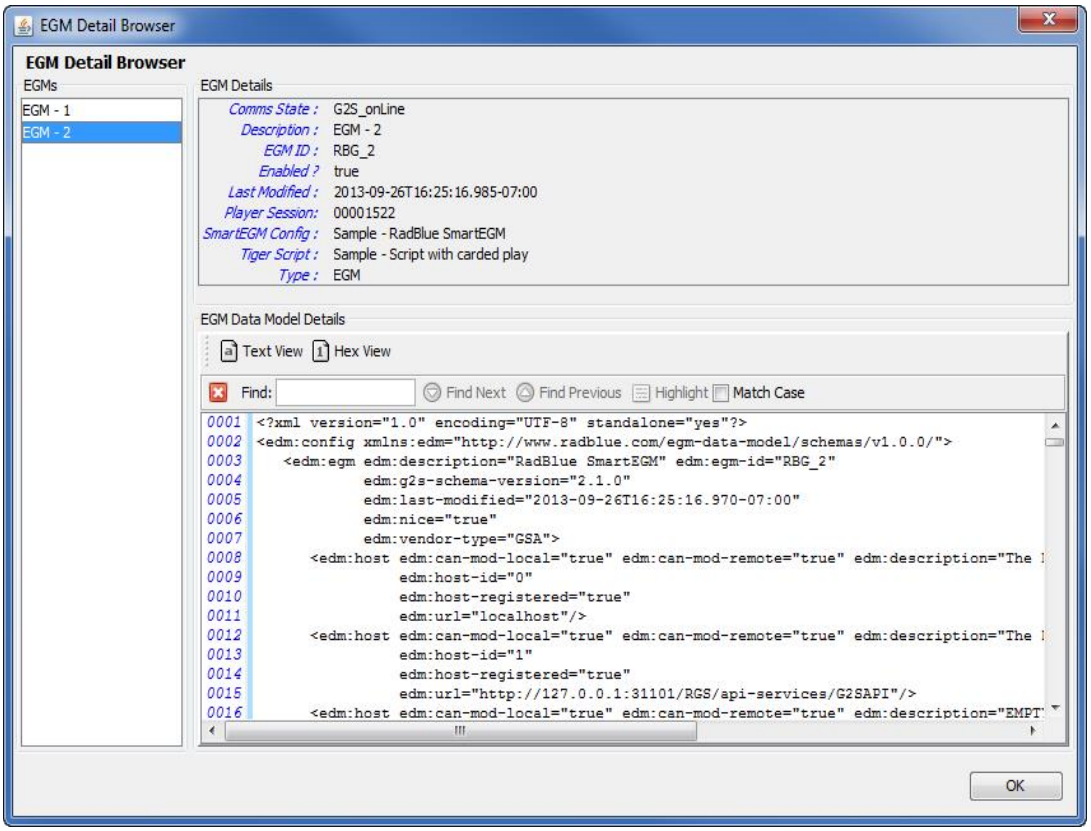
   or

   CTRL+click to select multiple EGMs to edit.

   or

   Click **Select All** if you want to edit all EGMs.

3. Click **View**.

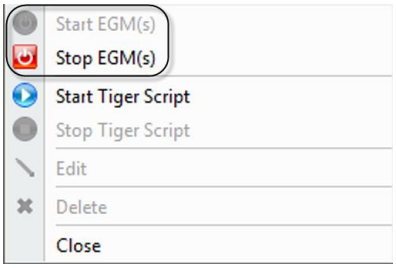


4. Review the details of the selected EGM.
5. To view the EGM data model in the EGM Data Model Details section
   - Click the **Hex View** option to view data model information in hexadecimal format.
   - Use the **Find** field search the data model for keywords. Click **Highlight** to highlight all instances of a keyword in the data model.
6. If you have selected multiple EGMs, click another EGM in the EGMs list to view it.
7. Click **OK** to close the EGM Detail Browser.

Start or Stop an Individual EGM

Once you have started the RLT engine by clicking **Start Engine** under **RLT Controls**, you can use the right-click option for any EGM to start or stop that EGM. If the RLT engine hasn't been started, the **start EGM(s)** and **stop EGM(s)** right-click options are disabled.

1. From the **Load Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. Click to select the EGM you want to start or stop.
3. Right-click the selected EGM.



4. If the EGM is running, click **Stop EGM(s)**; If the EGM is stopped, click **Start EGM(s)**.

Start or Stop a Tiger Script

Once you have started the RLT engine by clicking **Start Engine** under **RLT Controls**, you can start all or some of the Tiger Scripts associated with the EGMs.

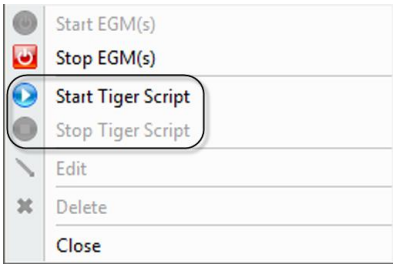
1. From the **Load Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. If you want to start all Tiger Scripts for all EGMs, click **Start All Tiger Scripts** under **RLT Controls**.

   or

   To start or stop the Tiger Script of an individual EGM:

   a. click to select the EGM with the Tiger Script you want to start or stop (or CTRL+click to select multiple EGMs).

   b. Right-click the selected EGM.

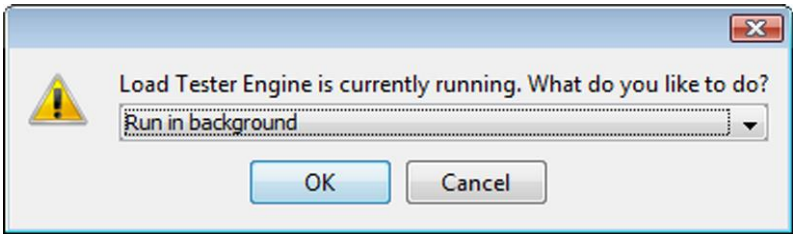


   c. If the Tiger Script is stopped, click **Start Tiger Script**; If the Tiger Script is stopped, click **Stop Tiger Script**.

Run EGMs in the Background

If you currently have EGMs running, you can close RLT and keep those EGMs running in the background. You can launch the RLT user interface at any time by clicking the RLT icon in the system tray.

1. From the **Load Tester** menu, click **EGMs**.
2. Under **Manage EGMs**, verify that the EGMs you want to run are currently running:
   - **Enabled** is set to **true**.
   - **Comms State** is set to **G2S_online**.
3. Click the "x" in the top right-hand corner of the RLT screen to close RLT.



4. Verify that **Run in Background** is selected.

   or

   Click the drop-down arrow, and select **Exit** to close RLT without running EGMs in the background. In this case, the running EGMs will have a communication state of "lost" when you restart RLT.

5. Click **OK**.

   An RLT icon is created in the system tray if you choose to run RLT in the background.

6. To launch RLT user interface, click the RLT icon in the system tray.

Delete EGMs from RLT

You can delete EGMs from RLT through the EGMs option.

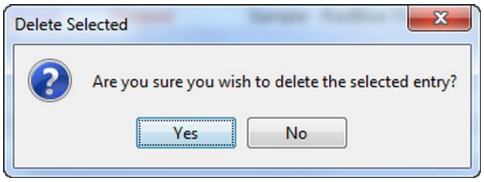
1. From the **Load Tester** menu, click **EGMs**. The **Manage EGMs** screen displays in the content area to the right.
2. Click to select the EGM you want to delete.

   or

   CTRL+click to select multiple EGMs to delete.

   or

   Click **Select All** if you want to delete all EGMs.
3. Click **Delete**.



4. Click **Yes** to delete the selected EGM(s) from RLT.

About SmartEGM Templates

From Manage SmartEGM Templates, you can import SmartEGM templates for use with RLT EGMs. For each available template, you can change the file description, modify host system information and view the devices. A sample SmartEGM template is available by default when you install RLT, so you can get up and running quickly.

Note that, while you can view host or device XML, you cannot edit SmartEGM templates through RLT or the viewer.

Import a SmartEGM Template into RLT

1. Click **Add**.
2. Click the Config File icon, and browse to the file you want to import.
3. Type a description for the SmartEGM Template you are importing.
4. Click **Open**.
5. Click **Save**.

Edit a SmartEGM Template's Host Information

1. Click to select the SmartEGM Template you want to modify. The associated host and device information displays in the panel below the SmartEGM template list.
2. On the **Host** tab, click to select the host you want to change.
3. Click **Edit Host**.
4. Modify the following fields as needed:
   - **Host ID** – Select or type a new identifier for the selected host.
   - **Description** – Type a new description for the selected host.
   - **Host URL** – Type a new network address for the selected host.
5. Click **Save**.
   The imported SmartEGM template is added to the Manage SmartEGM templates list.

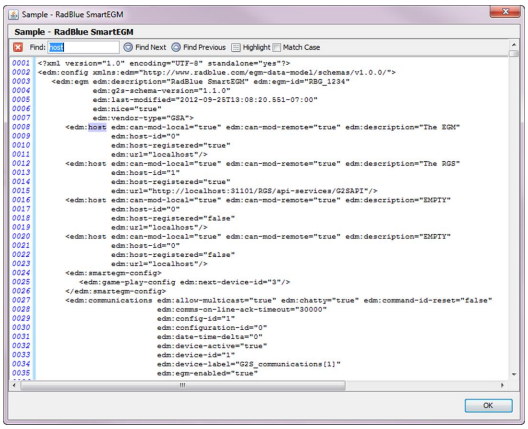
View Supported Devices for a SmartEGM Template

1. Click to select the SmartEGM Template you want to view.
2. Click the **Devices** tab displayed in the panel below the SmartEGM Template list to see all supported devices for the selected template. Note that the Device list is *read-only*.

View XML for a SmartEGM Template

From SmartEGM Templates, you can view the XML for the host or device of the selected SmartEGM template. Note that the XML cannot be modified through the XML viewer.

1. From the Manage menu, click **SmartEGM Templates**.
2. Click to select the SmartEGM Template you want to view from the **Manage SmartEGM Templates** list.
3. Click **View**.



The **Find** option lets you search for text strings and keywords within the XML file. Simply click in the **Find** text box and start typing. The tool instantly jumps to the first match of the entered string. This feature is handy when you want to find specific data in the SmartEGM Template.

- Use **Find Next** and **Find Previous** to move to the next or previous match of the entered string.
- Click **Highlight** to highlight all instances of the text string or keyword in the file.
- Select **Match Case** if you want to find only a text string or keyword with a specific case (capital or lower case letters).

4. Click **OK** to close the XML viewer.

Change a SmartEGM Template Description

1. Click to select the SmartEGM template you want to modify. A Description field displays below the SmartEGM template list.
2. Type a new description for the selected template in the Description field.
3. Click Save Description.

Delete a SmartEGM Template

1. Click to select the SmartEGM template you want to delete.
2. Click Delete.
3. When prompted, click Yes to delete the template from RLT.

About Tiger Scripts

From Manage Tiger Scripts, you can import Tiger scripts into RLT for use with RLT EGMs as well as view the contents of Tiger scripts.

Note that you cannot edit a Tiger script through RLT. For detailed information on creating and modifying Tiger scripts, see the Tiger Scripting Reference.

Import a Tiger Script

1. Click **Add**.
2. Click the **Script File** folder icon, and browse to the Tiger script file you want to import.
3. Type a description for the Tiger script you are importing.
4. Click **Save**.
   The imported Tiger script is added to the Manage Tiger Scripts list.

View Tiger Script Information

You can a Tiger script's metadata, Tiger verbs and source code by simply clicking to select the script you want to view.

- The **Metadata** tab contains information about the script such as the author, when it was created and the G2S protocol version it uses.
- The **Tiger Verbs** tab displays each of the Tiger verbs contained in the script, in the order they occur.
- The **Tiger Source** tab displays the Tiger script source code. Use the Find option to locate specific items in the script.

Delete a Tiger Script

1. Click to select the Tiger script you want to delete.
2. Click Delete.
3. When prompted, click Yes to delete the script from RLT.

About Game Outcome

Game outcome is an extension to G2S protocol's gamePlay class. This extension provides host systems with detailed game play information by reporting the specific details of game play and game outcome. The protocol extension , *IGT gamePlay Class Game Outcome Extension*, defines support for poker games. In addition to poker, RadBlue-defined keno and reel games have been implemented in the tool.

Depending on the game type, game outcome reports:

- initial and final cards
- keno draws, keno picks and keno hits
- reel lines played and reel position symbols

The events related to game outcome are:

- IGT_GPE101 - Initial Outcome Logged (*poker only*)
- IGT_GPE102 - Initial Outcome Updated (*poker only*)
- IGT_GPE103 - Final Outcome Logged

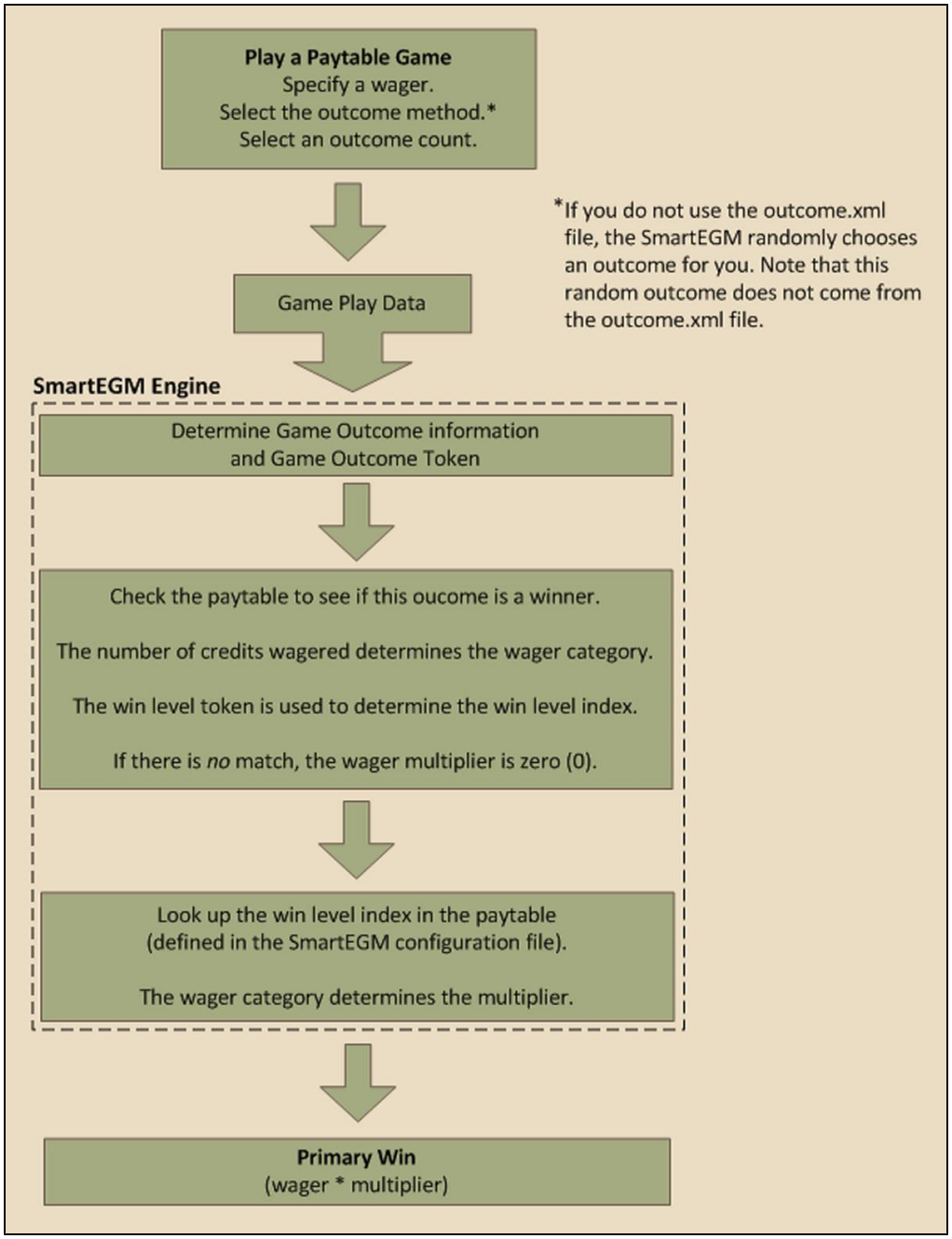
The content of each of these events, specifically, the `outcomeLog` element, depends on the type of game played. See the *IGT gamePlay Class Game Outcome Extension* document and Extensions to the G2S gameplay.outcomeLog Command for detailed information.

The SmartEGM can be configured to select a specific game outcome for repeatable host testing (using a seed number and the outcome.xml file) as well as random game outcomes within a set of outcomes that you specify.

You can specify the outcome of games using the outcome.xml file. This file lets you define the specific first outcome for each game that is played. You can also set the weight of each game outcome, which influences how often a particular game outcome is selected by the SmartEGM. If you choose not to use the outcome.xml file, the SmartEGM randomly selects the outcome when the Play Paytable Game player verb is used.

How Game Outcome Works

When a paytable game is played through a Tiger script, game options go to the SmartEGM. The SmartEGM then calculates the game outcome. If you do not require a specific game outcome (the outcome.xml file is *not* used), the SmartEGM randomly determines a final game outcome.

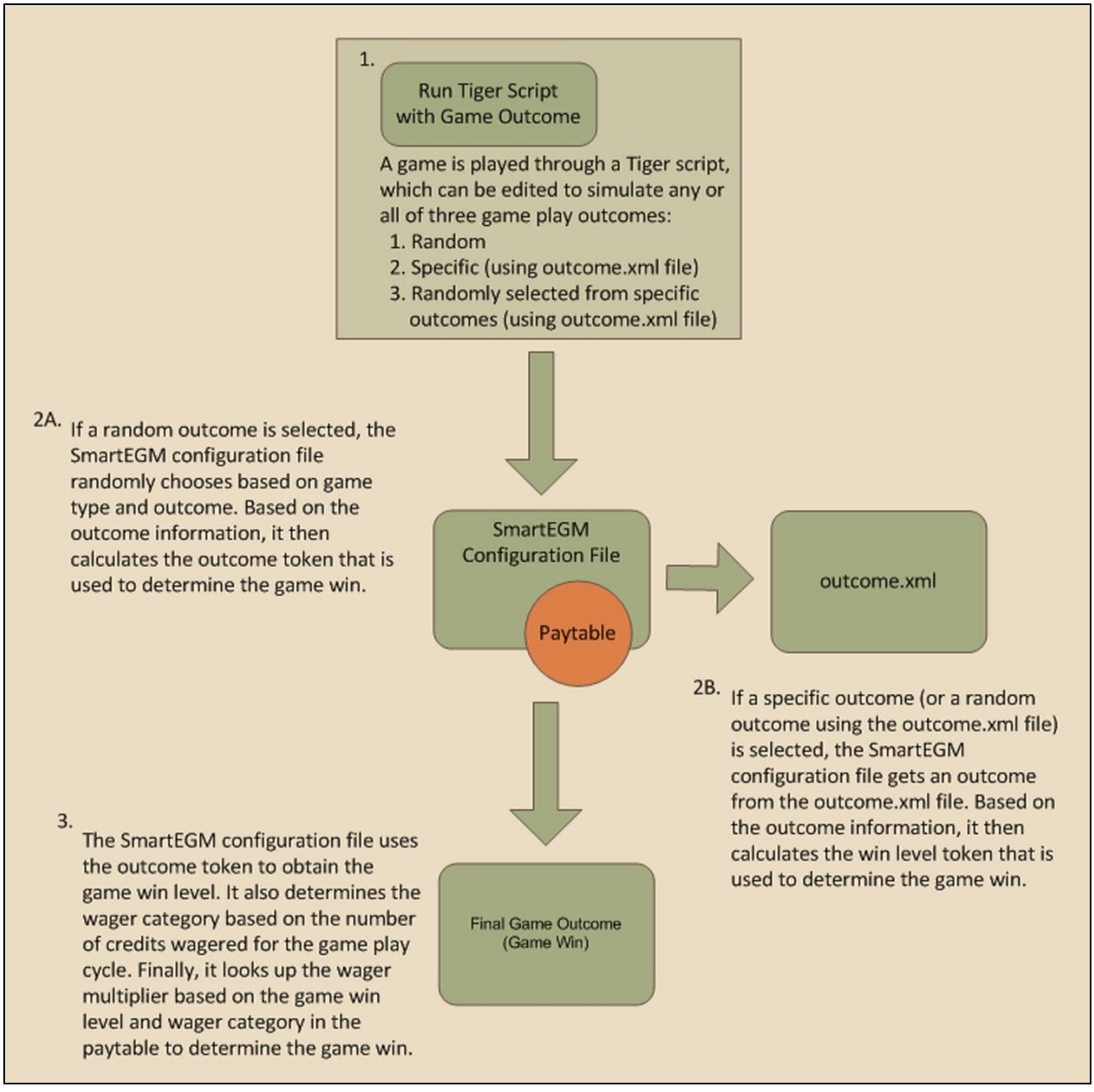


The final game outcome is sent in the `IGT_GPE103` (Final Outcome Logged) event, which you can view through the Message Transcript.

If you want specific game outcomes, you have two choices:

1. Use the same seed value along with the outcome.xml file for each game played. This will give you repeatable results because the random number generator in the SmartEGM will start at the same value each time it selects the final outcome, selecting outcomes from the same game outcome choices.
2. Use the outcome.xml file to pre-define weighted game results. Game outcomes are randomly pulled from this file.

Game Outcome Flow in RLT



Example Game Outcome Flows

The following game outcome examples walk you through the flow of messages during game outcome play. In addition, each example shows you the related messages that are sent and received by the tool, and displayed in the Message Transcript (denoted as *Transcript*).

Play a Paytable Game with a Random Outcome

1. A paytable game is played with a random outcome selected. **Game play data** is sent to the SmartEGM.

   Transcript: G2S_GPE103 - Primary Game Started

2. Using the game play data, the SmartEGM calculates a random **game outcome**. For example, the SmartEGM calculates the **game outcome** is **three cherries**.
3. The SmartEGM then determines the **game outcome token** that represents that **game outcome**. For example, the token would be **3-Cherries**.
4. The SmartEGM then looks up the **win level index** that is associated with the **game outcome token** from the smartegm-configuration.xml file. Note: that if the **game outcome token** is not in the **win level index** the game is considered a lost game and will have a **wager multiplier** of zero (0).
5. Based on the number of credits wagered, the SmartEGM then determines the appropriate **wager category** for this game play cycle.
6. The SmartEGM then determines the **wager multiplier** for the **win level** and **wager category**. If there is no paytable entry for the specific **win level** and **wager category** the game is considered a lost game and will have a **wager multiplier** of zero (0).

   Transcript: IGT_GPE101 - Initial Outcome Logged (poker only)

   Transcript: IGT_GPE102 - Initial Outcome Updated (poker only)

   Transcript: IGT_GPE103 - Final Outcome Logged

7. The initial game win is calculated by multiplying the wager by the **wager multiplier**.
8. The SmartEGM continues the rest of the game play cycle as described in the G2S protocol.

   Transcript: G2S_GPE105 - Primary Game Ended

Play a Paytable Game with a Specific Outcome

1. A paytable game is played with a poker outcome selected. **Game play data** is sent to the SmartEGM.

   Transcript: G2S_GPE103 - Primary Game Started

2. Using the **game play data**, the SmartEGM retrieves the **game outcome** from the outcome.xml file.
3. Since this is a poker game, the Smart EGM determines if the initial poker hand is a winning hand. It does this by taking the initial hand and determining the **game outcome token** for this hand. If the **game outcome token** is associated with a win level in the smartegm-config.xml file, the initial hand is considered a winning hand and the attributes in the `IGT_GPE101` are updated appropriately.

   Transcript: IGT_GPE101 - Initial Outcome Logged

4. The SmartEGM then determines which cards must be discarded by comparing the initial hand to the final hand. If the card from the first hand is present in the final hand then that card is not discarded, otherwise the card is discarded.

   Transcript: IGT_GPE102 - Initial Outcome Updated

5. The SmartEGM then determines the **game outcome token** for the final **game outcome**.
6. The SmartEGM then looks up the **win level index** that is associated with the **game outcome token** from the smartegm-configuration.xml file. Note: that if the **game outcome token** is not in the **win level index** the game is considered a lost game and will have a **wager multiplier** of zero (0).
7. Based on the number of credits wagered, the SmartEGM then determines the appropriate **wager category** for this game play cycle.
8. The SmartEGM then determines the **wager multiplier** for the **win level** and **wager category**. If there is no paytable entry for the specific **win level** and **wager category** the game is considered a lost game and will have a **wager multiplier** of zero (0).

   Transcript: IGT_GPE103 - Final Outcome Logged

9. The initial game win is calculated by multiplying the wager by the **wager multiplier**.

Game Outcome Configuration Overview

To use game outcome configuration overview:

1. Configure paytable information in the SmartEGM configuration file.
2. Define game outcomes (*optional*).
3. Configure Tiger scripting for automated testing.

Configure Paytable Information in the SmartEGM Configuration File

To configure a paytable in the SmartEGM configuration file, you must edit the XML file.

> **Note:** The following instructions were completed using a Windows operating system. The procedure for Linux operating systems will vary.

1. Navigate to:[**installation directory**] > **sample** > **smart-egm**
2. Right-click **smartegm-config-gsa.xml**, and select **Edit**.
3. If you want to create a *new* SmartEGM configuration file for game outcome, go to **File** > **Save As**, rename the file (for example, **smart-egm-gameOutcome.xml**) and click **Save**.
4. To configure game outcome in the SmartEGM configuration file, modify the wager-categories element, win-levels element and paytable element for each gamePlay device. (See "SmartEGM Configuration File Game Outcome Elements and Attributes" on page 46
5. Save and close the file.
6. Load the updated file into the tool.

Example SmartEGM Configuration - Game Outcome

```
<!-- The Second G2S GAME PLAY Device -->
```

The first thing you must do when adding game outcome information to a gamePlay device in the SmartEGM configuration file is to define the game (`theme-type`). Valid theme types are `IGT_poker`, `IGT_reels`, `IGT_keno` and `IGT_none`.

```
<edm:game-play edm:device-id="2" edm:device-active="true" edm:configuration-id="0" edm:host-
enabled="true" edm:egm-enabled="true" edm:egm-locked="false" edm:host-locked="false"
edm:owner-id="1" edm:config-id="1" edm:vendor-id="RBG"
edm:product-id="RBG_SweatT-0012" edm:release-number="RBG_ST0099" edm:vendor-name="Radical
Blue Gaming" edm:product-name="RBG_SweatingMightily12" edm:serial-number="RBG_00000123"
edm:secondary-game-enabled="false" edm:theme-type="IGT_poker">
```

Next, you must define the wager categories associated with the game play device.

```
<edm:wager-categories>
              <edm:wager-category edm:min-wager-credits="1" edm:max-wager-credits="4"
edm:theoretical-payback-percentage="9473"
                            edm:category-name="RBG_WagerCat1" />
              <edm:wager-category edm:min-wager-credits="5" edm:max-wager-credits="1000"
edm:theoretical-payback-percentage="9610"
                            edm:category-name="RBG_WagerCat2" />
 </edm:wager-categories>
```

Next, you must define the win levels for the game play device. For each win-level, you specify the win-level-token that will be used to determine the win level index.

```
<edm:win-levels edm:win-level-matcher="Default">
                          <edm:win-level edm:index="1" edm:win-level-combo="Royal Flush"
edm:progressive-allowed="true" edm:win-level
                                        odds="40390" edm:win-level-token="PokerRoyalFlush" />
                          <edm:win-level edm:index="2" edm:win-level-combo="Straight Flush"
edm:progressive-allowed="true" edm:win-level
                                        odds="9148" edm:win-level-token="PokerStraightFlush"
/>
                          <edm:win-level edm:index="3" edm:win-level-combo="Four of a Kind"
edm:progressive-allowed="false" edm:win-level
                                       odds="423" edm:win-level-token="PokerFourOfAKind" />
                           <edm:win-level edm:index="4" edm:win-level-combo="Full House"
edm:progressive-allowed="false" edm:win-level-odds="86"
                                       edm:win-level-token="PokerFullHouse" />
                           <edm:win-level edm:index="5" edm:win-level-combo="Flush"
edm:progressive-allowed="false" edm:win-level-odds="90"
                                       edm:win-level-token="PokerFlush" />
                           <edm:win-level edm:index="6" edm:win-level-combo="Straight"
edm:progressive-allowed="false" edm:win-level-odds="89"
                                       edm:win-level-token="PokerStraight" />
```

```
                        <edm:win-level edm:index="7" edm:win-level-combo="Three of a Kind"
edm:progressive-allowed="false" edm:win-level
                                    odds="13" edm:win-level-token="PokerThreeOfAKind" />
                        <edm:win-level edm:index="8" edm:win-level-combo="Two Pairs"
edm:progressive-allowed="false" edm:win-level-odds="7"
                                    edm:win-level-token="PokerTwoPairs" />
                        <edm:win-level edm:index="9" edm:win-level-combo="Jacks or Better"
edm:progressive-allowed="false" edm:win-level
                                     odds="4" edm:win-level-token="PokerJacksOrBetter" />
</edm:win-levels>
```

Finally, you must associate the wager multiplier to a wager category within a win level, using the win level index.

```
            <edm:paytable>
                      <edm:paytable-win-level edm:win-level-index="1">
                                <edm:paytable-wager-category edm:category-name="RBG_
WagerCat1" edm:wager-multiplier="250" />
                                <edm:paytable-wager-category edm:category-name="RBG_
WagerCat2" edm:wager-multiplier="800" />
                      </edm:paytable-win-level>
. . .
                         <edm:paytable-win-level edm:win-level-index="9">
                                <edm:paytable-wager-category edm:category-name="RBG_
WagerCat1" edm:wager-multiplier="1" />
                                <edm:paytable-wager-category edm:category-name="RBG_
WagerCat2" edm:wager-multiplier="1" />
                         </edm:paytable-win-level>
             </edm:paytable>
</edm:game-play>
```

SmartEGM Configuration File Game Outcome Elements and Attributes

Use the following elements and attributes to add game outcome information to the SmartEGM configuration file.

wager-categories Element

This element defines the wager category table for each gamePlay device.

wager-categories Elements

Element	Restrictions	Description
wager-category	minOcc: 1 maxOcc: ∞	Contains a wager category that is supported by the gamePlay device.

wager-category Element

This element defines a wager category by specifying the category name, wager credit limits and theoretical payback percentage.

wager-category Attributes

Attribute	Restrictions	Description
category-name	type: string use: required	Wager category identifier.
max-wager-credits	type: integer use: required	Maximum wager credits for the selected wager category.
min-wager-credits	type: integer use: required	Minimum wager credits for the selected wager category.
theoretical-payback-percentage	type: decimal use: required	Theoretical payback percentage of associated wager category.

win-levels Element

This element defines the win level table for each gamePlay device.

win-levels Attributes

Attribute	Restrictions	Description
win-level-matcher	type: string use: optional default: "Default"	Unique identifier that specifies the code in the SmartEGM that determines the game outcome token. Currently only Default is defined.

win-levels Elements

Elements	Restrictions	Description
win-level	minOcc: 1 maxOcc: ∞	Contains a win level that is supported by the game play device.

win-level Element

This element defines each win level in the gamePlay device.

win-level Attributes

Attribute	Restrictions	Description
index	type: positive integer use: required	Unique paytable index for the win level.
win-level-combo	type: string use: optional default: " "	Description of paytable win level.
progressive-allowed	type: boolean use: optional default: "false"	Indicates whether the paytable win level can be assigned to a progressive game play device.
win-level-odds	type: positive integer use: optional default: "0"	Theoretical weight of associated win level.
win-level-token	type: string use: optional default: "Default"	Game outcome token that is associated with this win level.

paytable Element

This element defines the paytable for the gamePlay device.

paytable Elements

Element	Restrictions	Description
paytable-win-level	minOcc: 1 maxOcc: ∞	Paytable entry.

paytable-win-level Element

This element associates a wager multiplier for each win level index and wager category. There must be a *paytable-wager-category* attribute for each wager category specified in the wager category table for this gamePlay device.

paytable-win-level Attributes

Attribute	Restrictions	Description
win-level-index	type: positive integer use: required	Unique identifier within a paytable for a specific prize level.

paytable-win-level Elements

Element	Restrictions	Description
pyatable-wager-category	minOcc: 1 maOcc: ∞	Associates win level index and wager category to a wager multiplier.

paytable-wager-category Element

This child element of the `paytable-win-level` element specifies the wager multiplier for the win level and wager categories.

paytable-wager-category Attributes

Element	Restrictions	Description
category-name	type: string use: required	Wager category name.
wager-multiplier	type: positive integer use: required	Number that is multiplied by the wager to determine the game's initial win.

SmartEGM Paytables

SmartEGM paytables provide possible game outcomes for default SmartEGM games. The following paytables list available game outcomes for keno, reel games and poker games.

Keno Default Outcomes

The default keno game is based on an 80-ball game, with 10 player picks for each game play and 20 draws for each game outcome.

Win Level Token	Description
HIT-0	There were no player picks that matched the drawn balls.
HIT-1	There was one player pick that matched the drawn balls.
HIT-2	There were two player picks that matched the drawn balls.
HIT-3	There were three player picks that matched the drawn balls.
HIT-4	There were four player picks that matched the drawn balls.
HIT-5	There were five player picks that matched the drawn balls.
HIT-6	There were six player picks that matched the drawn balls.
HIT-7	There were seven player picks that matched the drawn balls.
HIT-8	There were eight player picks that matched the drawn balls.
HIT-9	There were nine player picks that matched the drawn balls.
HIT-10	There were ten player picks that matched the drawn balls.

Reels Game Default Outcomes

Win Level Token	Description
1-Cherry	First reel stop has a cherry symbol.
2-Cherries	First and second reel stops have cherry symbols.
3-Cherries	All three reel stops have cherry symbols.
2-Single-Bars	First and second reel stops have single bar symbols.
3-Single-Bars	All three reel stops have double bar symbols.
2-Double-Bars	First and second reel stops have triple bar symbols.
3-Triple-Bars	All three reel stops have single 7 symbols.
3-Single-7s	All three reel stops have single 7 symbols.
3-Double-7s	All three reel stops have double 7 symbols.

Win Level Token	Description
3-Triple-Bars	All three reel stops have triple 7 symbols.
3-Any-Bars	All three reel stops have a bar symbol.

Poker Default Outcomes

Win Level Token	Description
PokerRoyalFlush	Royal Flush
PokerStraightFlush	Straight Flush
PokerFourOfAKind	Four of a Kind
PokerFullHouse	Full House
PokerFlush	Flush
PokerStraight	Straight
PokerThreeOfAKind	Three of a Kind
PokerTwoPairs	Two Pairs
PokerJacksOrBetter	Jacks or Better

Configure the outcome.xml File

The outcome.xml file lets you specify available outcomes for poker, keno and reel games.

To modify the outcome records used by the RLT engine:

1. Navigate to the **RLT installation directory**, and open the **conf** folder
2. Right-click the **outcome.xml** file, select **Open With...** > **Notepad**.
3. Use the outcome.xml elements and attributes as well as the game outcome example to modify the outcome.xml file.
4. **Save** and **close** the file.
5. Open **RLT**.
6. From the **RLT Community**, click **Tiger Scripts**.
7. Click the **Game Outcome** tab.
8. To import the outcome.xml file.
   a. Click **Import**.
   b. Click the **Browse** icon, and navigate to the file location.
   c. Select the file, and click **Open**.

   or

   If there is an existing outcome.xml file:

   a. Click+SHIFT to highlight all of the existing outcome.xml files (each theme is listed individually).
   b. Click **Delete**.
   c. Click **Import**.

   The new or updated file will be used the next time the RLT engine is started.

Example outcome.xml File

The following outcome.xml content shows outcome information for a poker, keno and reels games.

```
<?xml version="1.0" encoding="UTF-8"?>
<outcome-record-list xmlns="http://www.radblue.com/outcome">
  <outcome-record name="poker-001" weight="100">
    <poker>
      <initial-hand>
        <card suit="IGT_hearts" rank="IGT_ace" />
        <card suit="IGT_diamonds" rank="IGT_two" />
        <card suit="IGT_spades" rank="IGT_seven" />
        <card suit="IGT_clubs" rank="IGT_king" />
        <card suit="IGT_hearts" rank="IGT_king" />
      </initial-hand>
      <final-hand>
        <card suit="IGT_hearts" rank="IGT_ace" />
        <card suit="IGT_diamonds" rank="IGT_ace" />
        <card suit="IGT_clubs" rank="IGT_ace" />
        <card suit="IGT_clubs" rank="IGT_king" />
        <card suit="IGT_hearts" rank="IGT_king" />
      </final-hand>
    </poker>
  </outcome-record>
  <outcome-record name="keno-001" weight="100">
    <keno>
      <picks>
        <pick number="2" />
        <pick number="5" />
        <pick number="12" />
        <pick number="19" />
        <pick number="25" />
        <pick number="27" />
        <pick number="54" />
        <pick number="65" />
        <pick number="71" />
        <pick number="80" />
      </picks>
      <draws>
        <draw number="5" />
        <draw number="7" />
        <draw number="3" />
        <draw number="12" />
        <draw number="19" />
        <draw number="21" />
        <draw number="26" />
        <draw number="28" />
        <draw number="32" />
        <draw number="37" />
        <draw number="49" />
        <draw number="52" />
```

```
        <draw number="57" />
        <draw number="61" />
        <draw number="67" />
        <draw number="68" />
        <draw number="72" />
        <draw number="75" />
        <draw number="77" />
        <draw number="79" />
      </draws>
    </keno>
  </outcome-record>
  <outcome-record name="reels-001" weight="100">
    <reels>
      <pay-line>
        <stop symbol="RBG_cherry" />
        <stop symbol="RBG_bar" />
        <stop symbol="RBG_seven" />
      </pay-line>
    </reels>
  </outcome-record>
</outcome-record-list>
```

outcome.xml Elements and Attributes

outcome-record-list Element

This element is the root element for the outcome.xml file.

outcome-record-list Elements

Element	Restrictions	Description
outcome-record	minOcc=1 maxOcc=∞	An outcome record.

outcome-record Element

This element describes a single game outcome. The name attribute must be unique in the XML document.

outcome-record Attributes

Attribute	Restrictions	Description
name	type: string use: required	Thename of the outcome record.
weight	type: positive integer use: optional default: 1	The weight of this outcome record when randomlying selecting an outcome record.

outcome-record Elements

Element	Restrictions	Description
poker	minOcc=0 maxOcc=∞	Poker outcome.
keno	minOcc=0 maxOcc=∞	Keno outcome.
reels	minOcc=0 maxOcc=∞	Reels outcome.

poker Element

This element describes a poker game outcome.

poker Elements

Element	Restrictions	Description
initial-hand	minOcc=0 maxOcc=1	Initial poker hand.
final-hand	minOcc=1 maxOcc=1	Final poker hand.

initial-hand Element

This element describes the initial hand that is dealt to the player.

initial-hand Elements

Element	Restrictions	Description
card	minOcc=1 maxOcc=∞	A card in the initial hand.

card Element

This element describes a card.

card Attributes

Attribute	Restrictions	Description
suit	type: <lookup IGT's types>	Card's suit
rank	type: <lookup IGT's types>	Card's rank.

final-hand Element

This element describes the final hand that is dealt to the player. If a card in the final hand is identical to the initial hand then that card is held between the initial hand and the final hand.

final-hand Elements

Element	Restrictions	Description
card	minOcc=1 maxOcc=∞	A card in thefinal hand.

keno Element

This element describes a keno game outcome.

keno Elements

Element	Restrictions	Description
picks	minOcc=1 maxOcc=1	List of player's picks.
draws	minOcc=1 maxOcc=1	List of balls drawn by the game.

picks Element

This element lists the player's picks for the keno game.

picks Elements

Element	Restrictions	Description
pick	minOcc=1 maxOcc=∞	A player's pick.

pick Element

pick Attributes

Attribute	Restrictions	Description
number	type: positive integer	Number of the ball picked.

draws Element

This element lists the balls drawn for the keno game.

draws Elements

Element	Restrictions	Description
draw	minOcc=1 maxOcc=∞	A ball drawn by the keno game.

draw Element

This element describes a ball drawn by the keno game.

draw Attributes

Attribute	Restrictions	Description
number	type: positive integer	Number of the ball drawn.

reels Element

This element describes a reels game outcome.

reels Elements

Element	Restrictions	Description
pay-line	minOcc=1 maxOcc=1	Payline in the reel game.

pay-line Element

This element describes a pay line in a reels game.

pay-line Elements

Element	Restrictions	Description
stop	minOcc=1 maxOcc=∞	One reel stop on the reels game.

stopElement

This element describes a reel stop.

stop Attributes

Attribute	Restrictions	Description
symbol	type: SymbolTypes	Symbol show at this reel stop.

Add Game Outcome to a Tiger Script

To add game outcome to a Tiger script, configure the `tiger:Human.PlayPaytableGame` verb.

tiger:Human.playPaytableGame

The `tiger:Human.playPaytableGame` verb simulates game play using paytables to determine game outcomes. This verb is used primarily to test the Game Outcome extension of the G2S protocol. Attributes and elements related to the Game Outcome extension appear in **bold**.

Attributes

Attribute	Restrictions	Description
credits-to-wager-cashable	type: positive integer use: optional default: "0"	Number of cashable credits to wager.
credits-to-wager-non-cashable	type: positive integer use: optional default: "0"	Number of non-cashable credits to wager.
credits-to-wager-promo	type: positive integer use: optional default: "0"	Number of promotional credits to wager.
denom-id	type: long use: required	Identifier of the denomination to be wagered.
tiger:device-id	type: integer use: optional default: "-2"	Device Identifier. • A value of "-2" means the first gamePlay device. • A value of "-1" is legal, but should not be used because it will result in a failure.
final-hand-count	**type: positive integer use: optional default: "1"**	**Number of final hands to generate.**
handpay-action	type: tiger:KeyOffAction use: optional default: "HANDPAY"	Indicates how the handpay should be paid.
in-game-delay	type: positive integer use: optional default: "0"	Defines an in-game delay, in milliseconds, that is executed after `GPE101` and after `GPE109`. Use this delay to simulate complex game mechanics.

Attribute	Restrictions	Description
tiger:key-off-timeout	type: integer use: optional default: "0"	How long to wait for a `setRemoteKeyOff` command before timing out. If a time out occurs, the `handpay-action` value is used.
play-secondary-game-count	type: positive integer use: optional default: "0"	Number of double-or-nothing secondary games to play if the game's calculated initial win is not zero. The first secondary game takes the game's initial win value as the amount wagered. For all but the last secondary game, the secondary amount won will equal twice the amount wagered.
remote-key-off-timeout	type:positive integer use: optional default: "60000"	**DEPRECATED 03 OCT 2012 - Version 24** **Replaced by `tiger:key-off-timeout`** Indicates how long to wait for the remote key off.
seed	**type: integer use: optional default: "-1"**	**Seed for the random number generator. Specifying a seed number provides a consistent series of game outcomes.**
win-final-secondary-game	type: boolean use: optional default: "false"	Indicates whether the final secondary game is a win or a loss. If `win-final-secondary-game` is set to **false**, the final win is zero (0).
win-to-handpay	type: boolean use: optional default: "false"	Indicates whether the game win goes to a handpay.

Element

Note: If this element is not included, the SmartEGM will always create random outcomes.

Attribute	Restrictions	Description
outcome	**type: tiger:OutcomeRecordType minOccurs: 0 maxOccurs: 1**	**Indicates the use of the outcome.xml file to determine the game outcome.**

tiger:OutcomeRecordType Attributes

Attribute	Restrictions	Description
outcome-random	type: boolean use: optional default: "false"	Indicates whether the outcome record is selected randomly from the outcome.xml file. If this value is true then outcome-record-name is ignored.
outcome-record-name	type: string	Name of selected outcome record from the

Attribute	Restrictions	Description
	optional default: " "	outcome.xml file.

This verb may be included in: tiger:action, tiger:catch, , tiger:repeat, tiger:then, tiger:tiger and tiger:try.

Examples

1. **Poker game with a specified outcome.**

```
<tiger:Human.playPaytableGame
       tiger:credits-to-wager-cashable=”50”
       tiger:denom-id=”25000”
       tiger:device-id=”2”
       tiger:final-hand-count=”50”>
     <tiger:outcome tiger:outcome-record-name=”outcome-001” />
</tiger:Human.playPaytableGame>
```

   Wager $12.50 on game play device 2. The initial and first final hands are specified in the outcome-list.xml file under the name “outcome-001”. The second through fiftieth final hands are randomly generated from a standard 52 card deck where the initial cards have been removed. See discussion about final hand generation.

   The primary win is calculated by apportioning the total credits-to-wager by the number of outcome hands to each individual final hand and looking up the wager multiplier for each winning hand.

2. **Poker game with a randomly selected outcome.**

```
<tiger:Human.playPaytableGame
       tiger:credits-to-wager-cashable=”5”
       tiger:denom-id=”100000”
       tiger:device-id=”2” >
     <tiger:pokerGame tiger:outcome-random=”true” />
</tiger:Human.playPaytableGame>
```

   Wager $5.00 on game play device 2. The initial hand and final hand is randomly selected from the outcome-list.xml file. Each record in the outcome-list.xml file has a relative weight that is used in the selection of records from outcome-list.xml file.

   The primary win is calculated by determining if the final hand is a winning hand and looking up the wager multiplier for that hand.

3. **Poker game where no outcomes are specified.**

```
<tiger:Human.playPaytableGame
     tiger:credits-to-wager-cashable=”5”
     tiger:denom-id=”100000”
     tiger:device-id=”2”
     tiger:final-hand-count=”50” />
```

   Wager $5.00 on game play device 2. The initial hand is randomly generated from a standard 52 card deck. If the initial hand is a winning hand, those cards that make up the winning hand are held. The first through fiftieth final hands are then randomly generated. See discussion about final hand generation.

tiger:keyOffAction

Enumeration	Description
HANDPAY	Key off the game to a local handpay.
CREDIT	key off the game to the credit meter.
VOUCHER	Key off the game to a voucher.
WAT	Key off the game to a WAT account
REMOTE	Key off the game to a remote server. The verb waits for the remote-key-off-timeout period. If the remote key off is not received in that period, the verb fails.

Extensions to the G2S gameplay.outcomeLog Command

This section provides detailed information about each type of game (*theme types*). The theme types supported by the SmartEGM are:

- **Poker** - See *IGT gamePlay Class Game Outcome Extension*.
- Keno
- Reels

Keno Theme Type Definitions

The default keno game is based on an 80-ball game, with 10 player picks for each game play and 20 draws for each game outcome. A "hit" occurs when a number in the draw matches a player-selected ("pick") number. The total number of hits is then reported as the game outcome.

IGT_keno: outcomeLog Elements

Keno-Specific oucomeLog Elements

Element	Restrictions	Description
kenoGame	minOcc: 1 maxOcc: 1	Contains detailed information on keno game outcome.

kenoGame Elements

Element	Restrictions	Description
picks	minOcc: 1 maxOcc: 1	Contains information on the player picks.
outcomes	minOcc: 1 maxOcc: 1	Contains information on game draws.

picks Elements

Element	Restrictions	Description
pick	minOcc: 1 maxOcc: ∞	Represents a single number selection by a player in a keno game.

pick Attribute

Element	Restrictions	Description
number	type: positive integer use: required	One of 20 numbers selected (drawn) from an 80-ball keno game.

outcomes Elements

Element	Restrictions	Description
outcome	minOcc: 1 maxOcc: ∞	Result of game play.

outcome Attributes

Element	Restrictions	Description
winningCombination	type:boolean use: optional default: false	Set to true if this was a winning game. Otherwise, set to false.
winLevelIndex	type: t_winLevelIndex use: optional default: 0	Set to index of the unique win level if winning game. Otherwise, set to zero (0).
winLevelCombo	type: t_winLevelCombo use: optional default: ""	Set to the name of the unique win level if winning game. Otherwise, set to zero (0).

outcome Elements

Element	Restrictions	Description
draws	minOcc: 1 maxOcc: 1	Contains detailed information on keno game number draws.

draw Elements

Element	Restrictions	Description
draw	minOcc: 1 maxOcc: ∞	Single number drawn on an 80-ball keno game.

draw Elements

Element	Restrictions	Description
number	type: positive integer use: required	Set to the number of the drawn ball.
hit	type: boolean use: optional default: false	A match between the pick list and outcome list.

IGT_keno: Log Updates

Add a picks sub-element to kenoGame.

IGT_GPE103 - Add pick sub-element for each player's pick

Attribute	Set Value to . . .
number	A single number selection by a player in a keno game.

Add a kenoGame sub-element to outcomeLog.

IGT_GPE103 - Add outcome sub-element for each final game

Attribute	Set Value to . . .
winLevelCombo	Number of matches of picks and outcomes. This attribute is populated by the paytable.
winLevelIndex	Index of the unique win level if winning game. Otherwise, set to zero (0).
winningCombination	Set to true if this was a winning game. Otherwise, set to false.

Add a draws sub-element to the outcome element.

IGT_GPE103 - Add a draw sub-element for each draw in the game

Attribute	Set Value to . . .
number	Number of the drawn ball.
hit	Set to true if the drawn ball matches a player's pick.

Spinning Reel Theme Type Definitions

The default reel game is based on the reel stops for a 3-reel EGM.

IGT_reel: outcomeLog Elements

Reel-Specific outcomeLog Elements

Element	Restrictions	Description
reelPayLines	minOcc: 1 maxOcc: 1	Contains detailed information on spinning reel game outcome.

Elements

reelPayLines Elements

Element	Restrictions	Description
reelPayLine	minOcc: 1 maxOcc: ∞	Contains information on reel payline stops.

reelPayLine Attributes

Attribute	Restrictions	Description
winningPayline	type: boolean use: required	Set to true if this was a winning game. Otherwise, set to false.
winLevelIndex	type: t_winLevelIndex use: optional default: 0	Set to index of the unique win level if winning game. Otherwise, set to zero (0).
winLevelCombo	type: t_winLevelCombo use: optional default: ""	Set to the name of the name of the unique win level if winning game. Otherwise, set to "".

reelPayLine Elements

Element	Restrictions	Description
reelStop	minOcc: 1 maxOcc: ∞	Single reel stop.

reelStop Attributes

Attribute	Restriction	Description
position	type: positive integer use: required	Position of the reel.
symbol	type:symbolTypes use: required	Unique identifier.
wild	type:boolean use: optional default: false	Indicates whether the symbol is wild.

Reel Data Type

SymbolTypesBase Attrributes

Attribute	Restriction	Description
SymbolTypes	type: t_extensibleList enumeration: RBG_blank RBG_cherry RBG_bar RBG_doubleBar RBG_tripleBar RBG_seven RBG_doubleSeven RBG_tripleSeven	Possible symbol outcomes.

IGT_reels: Log Updates

IGT_GPE103 - Add reelPayLine sub-element for each final game

Attribute	Set Value to . . .
winPayline	Set to final position of symbols in relation to the payline.
winLevelIndex	Set to index of the unique win level if winning game. Otherwise, set to zero (0).
winningCombo	Set to true if this was a winning game. Otherwise, set to false.

IGT_GPE103 - For each reelPayLine element, add a sub-element for each reelStop in the game

Attribute	Restrictions	Description
position	type: positive Integer	Position of the reel.
symbol	type:rbgReels:SymbolType	Unique identifier.
win	type:boolean	Indicates whether the game is a winning game.

Configure RLT Engine Options

Engine Options allow you to configure various RLT options, including memory size, batch size, screen refresh rate and transport options. Engine options are grouped functionally into General and Transport. Click a tab to view the options for that group.

General

From the General tab on the Engine Options configuration screen, you can define the amount of memory allocated to RLT, whether RLT validates messages against the G2S schema, whether or not to automatically run Tiger scripts, the number of groups to divide EGMs into when bringing online and how often to refresh the RLT screen.

- **Auto Start Tiger Script** - Select **Yes** to automatically start each EGM's Tiger script when the startup algorithm completes.
- **Auto Snapshot Metrics** - Select **Yes** to automatically snapshot RLT metrics and export them to a file (requires restart). Use the **Snapshot Interval** to specify the time between snapshots. Metrics are sent to: [**Installation Directory**]**\logs\metric-snapshot-<COMPUTER NAME>.cvs**
- **Batch Delay** - Type or select the amount of time RLT waits between bringing each batch online.
- **End Player Sessions** - Select **Yes** to end player session on EGMs when a Tiger Script(s) stops. If this option is set to **No**, player sessions *may or may not* be ended when a Tiger Script(s) stops.
- **EGM Batch Size** - Type or select the number of EGMs to bring online at once.
- **Initial Memory Size (MB)** - Type or select the memory initially allocated to RLT. The default is **1024 MB**.
- **Max Memory Size (MB)** - Type or select the maximum allowable memory allocated to RLT. The default is **1024 MB**.
- **Persist EGM Data Model** - Select **true** to retain the current EGM data model when RLT is restarted.
- **Screen Refresh Rate** - Type or select how often you want the RLT screen to be refreshed.
- **Tiger Script Batch Size** - Type or select the number of Tiger scripts to bring online at once.
- **Validate Messages** - Select **Yes** if you want all messages sent and received by RLT to be validated against the associated schema.

Transport

From the Transport tab on the Engine Options configuration screen, you can define RGS settings related to message transport.

- **Bind To Address** - Click the drop-down arrow, and select the IP Address that you want RLT to use for communications.
- **Debug SSL Enabled** - Click the drop-down arrow, and select **Yes** to enable SSL logging. Enabling this option provides more SSL logging messages, allowing for additional troubleshooting if there are SSL communication issues.
- **GZIP Enabled** - Click the drop-down arrow, and select **Yes** to enable GZIP support.
- **Multiple IP Addresses** - Click to configure multiple EGM IP addresses. See Configure Multiple IP Addresses.
- **Protocol Type** - Click the drop-down arrow, and select **HTTP/SOAP** to use non-secure communications or **HTTPS/SOAP** to use secure (SSL) communications.
- **Remote Control URL** - URL of the RLT REST interface (Remote Control). This field is *read-only*. To copy the URL, highlight the address and type **CTRL+C**.
- **Scratch Pad URL** - URL of the example RLT REST interface. This field is *read-only*. To copy the URL, highlight the address and type **CTRL+C**.
- **SOAP Port** - Enter the port that you want RLT to use for communications. We recommend that you do not change the SOAP port unless you have a port conflict.
- **SOAP Timeout**- Enter the time using **mm:ss:sss** format or keep the 30-second default.
- **Use Content Length** - If **Yes**, the content length field created in HTTP message header is used. Enabling this option may impact performance.

About Multiple IP Addresses

The Manage Multiple IP Addresses option lets you create or delete multiple IP addresses to the Load Tester that are then assigned to the EGMs on each engine.

> **Warning!** You **must** disable the Dynamic Host Configuration Protocol (DHCP) on your computer before you can add multiple IP addresses and while you are using the multiple IP addresses feature.

The Load Tester assigns IP addresses to EGMs in the order EGMs request an IP address at start-up. The number of EGMs assigned to a single IP address depends on the number of IP addresses you create versus the number of EGM running in a given engine.

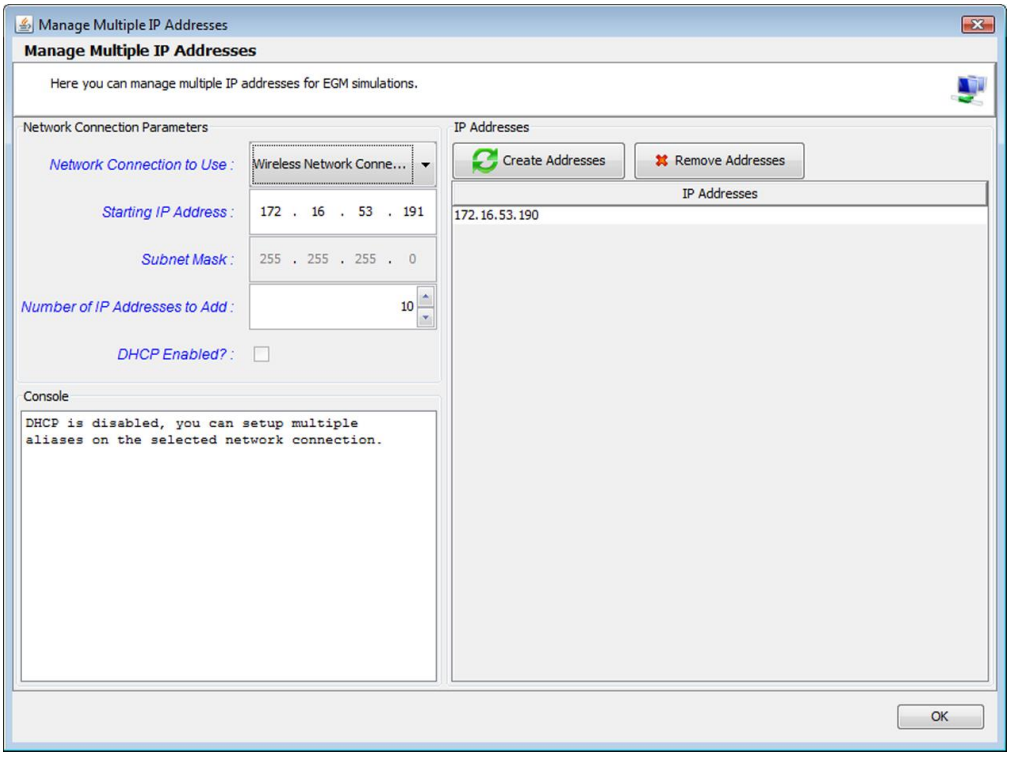
If you remove all 10 IP addresses, all EGMs on all engines would use the same IP address as soon as you restarted the Load Tester.

Note: To use the Manage Multiple IP Addresses option, your PC must have an active network connection.

Create Multiple IP Addresses

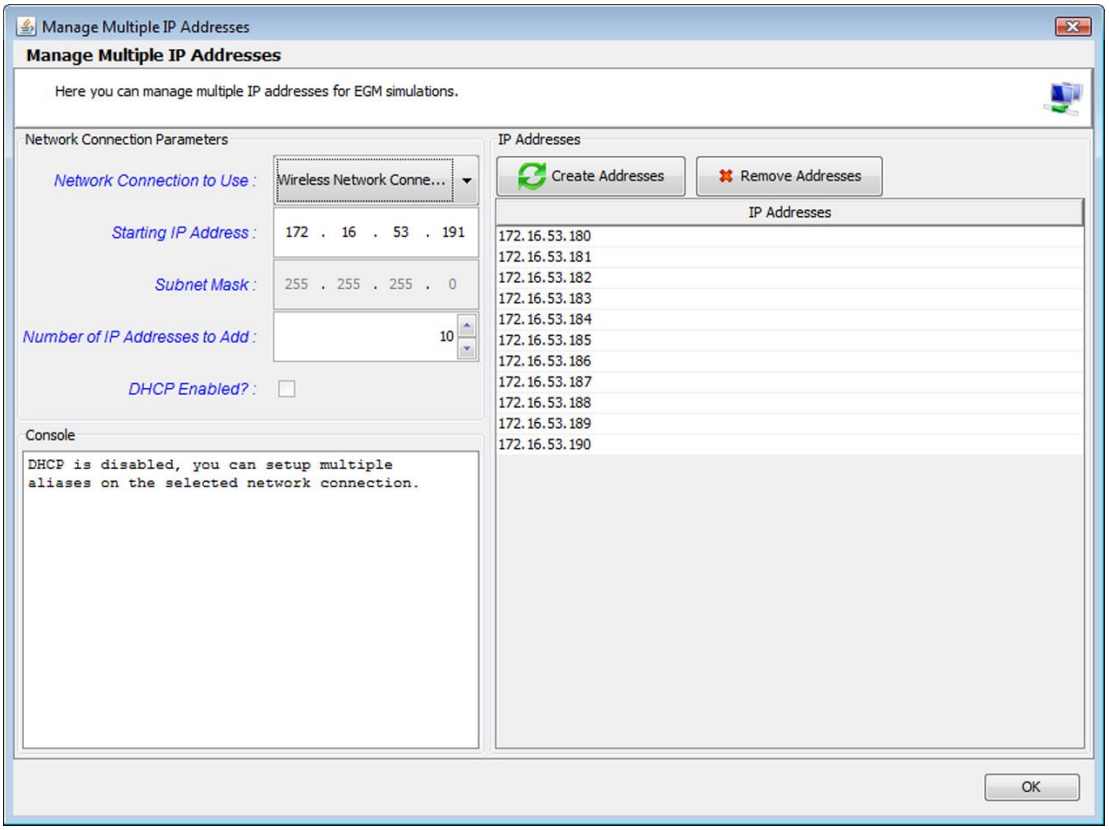
Use this procedure to add 1 to 200 IP addresses that are then randomly assigned to RLT EGMs.

1. Go to: **Tools** > **Configure** > **Engine Options**
2. Click **Manage Multiple IP Addresses**.



2. Configure the following **Network Connection Parameters**:
   - **Network Connection to Use** - Click the drop-down arrow, and select the type of network connection you want to use for the new IP addresses. The drop-down displays all IP addresses configured for all engine servers.
   - **Starting IP Address** - Enter the network address from which the new IP addresses will be generated.
   - **Subnet Mask** - Use the Subnet Mask to determine a valid IP address for your network.
   - **Number of IP Addresses to Add** - Enter the number of IP addresses you want to add (1-200).
   - **DHCP Enabled?** - For future use. DHCP is not available in this release of the Load Tester. If your computer uses DHCP, you must disable it. Contact your network administrator for assistance.

3. Click **Create Addresses**.



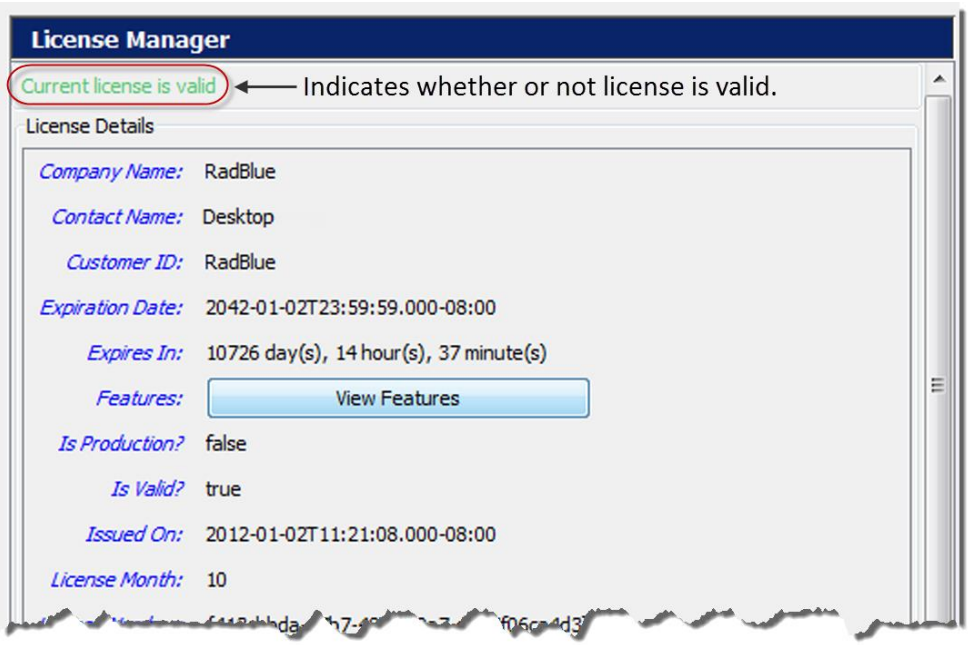
5. Click **OK** to return to the Load Tester Configuration screen.

Delete an IP Address

When using multiple IP addresses, you can delete existing IP addresses from RLT through the Manage IP Addresses screen. When you restart RLT, IP addresses are reallocated to EGMs based on the number of IP addresses remaining.

Configure License Manager Options

License Manager displays current licensing information, including the product features available under the license. The New License File option allows you to upload a new license file for the product.



License Details

- **Company Name** - Name of organization that purchased this license.
- **Contact Name** - Name of person license was issued to.
- **Customer ID** - Unique company identifier.
- **Expiration Date** - Date that tool becomes invalid.
- **Expires In** - Time left until license expiration.
- **Features** - Click **View Features** to see which features are enabled for your license.
- **Is Production?** - **True** indicates that the license is a fully licensed version.
- **Is Valid?** - **True** indicates that the license is valid; **False** indicates that the license is invalid.
- **Issued On** - Date of license issuance.
- **License Number** - Unique license identifier.
- **License Month** - Month that license expires.
- **License Year** - Year for which license is valid.
- **Load Message** - Status of license upload.

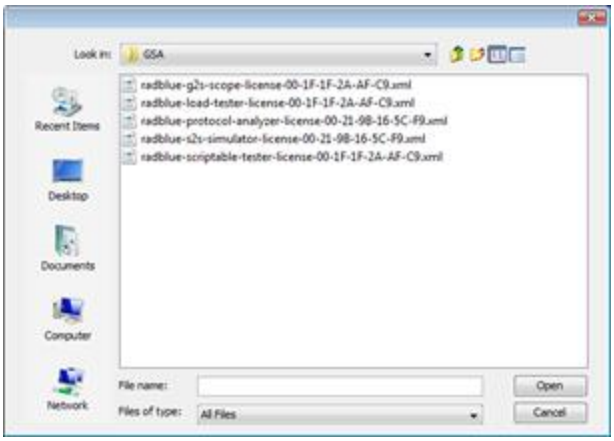
- **Location ID** - Location of purchasing organization.
- **MAC Address** - Physical address of computer on which the tool is installed.
- **Product Line Key** - Unique identifier of installed tool.
- **Product Name** - Name of licensed RadBlue product.

Load a New License File

To use the latest version of the tool, you may periodically need to update your license.

To load a new license:

1. Click the drop-down arrow.



2. Navigate to the new license file.
3. Highlight the new license file, and click **Open**.
4. Click **Apply** or **OK** to install the new license.

Configure Security Options

From Security Options, you can enable and configure Secure Socket Layer (SSL) encryption information.

- Enable and configure Online Certificate Status Protocol (OCSP) options
- Create and import signed certificates into the tool
- Manage installed keystore files

  Note: SSL configuration, including the Security Options screen, is not available in the Student Edition of the tool.

To use SSL security, you must select Enable SSL security control. You then have the option to select **Approve All Certificates** if you want to use SSL encryption, but are not concerned with the validity of the certificate authority.

If this option is cleared, the tool performs validity checking when an entity (for example, an EGM) initiates communications. The validity check includes:

- *Signed by trusted certificate authority?*
- *Is current time/date within the period of validity (effective and expired date)?*
- *Is issuer signature correct?*

When you make a change to the Security Options screen, you are prompted to restart the tool before your changes take effect.

Configure General Security Options

From the General tab, you can enable SSL in the tool, choose to approve all certificates, and configure OCSP options.

1. From **Tools** > **Configure** > **Security Options**.
2. Click **General**.

3. Select **Enable SSL security control** to use SSL encryption with the tool. This option must be selected to configure all additional security options.
4. Select **Use Minimum Security Environment** to enable minimum security option when testing. When you enable this option:
   - The **Transport Layer Security (TLS) 1.0** is the *only* supported protocol for client-side TLS sessions. Note that host-side sessions are not restricted.
   - The only supported cipher suite is SSL_RSA_WITH_3DES_EDE_CBC_SHA for both client-side and host-side TLS sessions.
5. Select **Require Client Certificate** if the other endpoint must have a certificate or it fails authentication. If this option is cleared, the other endpoint is not asked to send its client certificate. By default, this option is selected.
6. Enable and configure OCSP options.
7. Once all security options have been configured, click **OK**.

Enable and Configure OCSP

You can configure the Online Certificate Status Protocol (OCSP) options to check to see whether the certificate has been revoked.

If a certificate does not pass any validity check, an error is generated and the attempted connection will fail. You can view the error in the debug log.

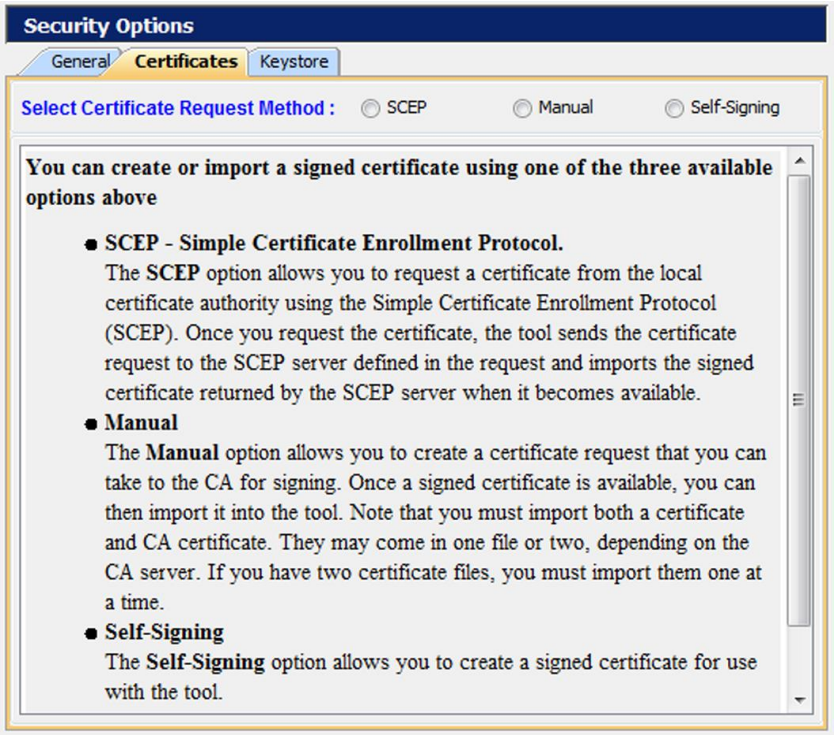
When you enable OCSP, you must configure the following options:

- **OCSP Server Location** - Type the URL location of the OCSP responder.
- **OCSP Server Considered Offline After (gsaOO) *x* Minutes** - Type or select the minimum period, in minutes, that the tool will attempt to authenticate a certificate from an OCSP server. Zero (0) disables this setting.
- **Re-Authenticate Certificate Every (gsaOR) *x* Minutes** - Type or select the maximum time, in minutes, that the tool can use a certificate without re-authenticating it.
- **Accept Previously Good Certificate for (gsaOA) *x* Minutes** - Type or select the maximum time, in minutes, that the tool can use a good certificate when OCSP servers are offline. Note that the gsaOA value should be greater period than the gsaOR value; The difference between gsaOR and gsaOA is the “accept offline” period.

Create or Import a Signed Certificate

You can create or import a signed certificate using one of three available options:

- Use SCEP to Request a Certificate
- Load a Manual Certificate
- Load a Self-Signing Certificate



To access the certificate options:

1. From the menu bar, click **Tools**.
2. Select **Configuration**.
3. Click **Security Options** to display the Security Options screen.
4. Click the **Certificates** tab.

Use SCEP to Request a Certificate

The SCEP option lets you request a certificate from the local certificate authority using the Simple Certificate Enrollment Protocol (SCEP). Once you request the certificate, the tool sends the certificate request to the SCEP server defined in the request and imports the signed certificate returned by the SCEP server when it becomes available.

1. From **Tools** > **Configure** > **Security Options**.
2. Click **Certificates**.
3. Select **SCEP** as the **Certificate Request Method**.
4. Configure SCEP options as required. The values related to the certificate authority (CA) are available from the CA provider.
   - **SCEP Server Location** - Type the network location of the certificate authority to which you want to send your certificate request.
   - **Pre-Shared Secret Enabled?** - Select if you want to include a pre-shared secret in the certificate request.
   - **Pre-Shared Secret** - Type the pre-shared password that you want to include in the certificate request.
   - **User Name Enabled?** - Select to include the user name in the certificate authority request.
   - **User Name** - Type the user name used by the certificate manager. Depending on your SCEP implementation, the user name may be included in the transaction ID or as part of the Certificate Signing Request (CSR) as the Common Name (CN).
   - **Use User Name as Common Name?** - Select if the user name is the same as the common name.
   - **Common Name** - Type the tool's common name. In the case of an EGM, the common name would be the EGM identifier. The default is **-1**.
   - **CA Identity Enabled?** - Select to include the identifier of the certificate authority in the request.
   - **CA Identity** - Type the identifier of the certificate authority to which the request will be sent.
   - **Entity Type** - Click the drop-down arrow, and select the role of the tool: G2S Host, G2S EGM Proxy, G2S EGM, Other G2S.
   - **Organization Unit** - Type the organizational unit (role) of the tool: G2S_host, G2S_egmProxy, G2S_egm, or Other G2S. By default, this field is populated with a value that corresponds to the Entity Type.
   - **Key Size** - Click the drop-down arrow, and select the size of the key pair supported by your network environment. (1024 is generally the most commonly accepted key size.)

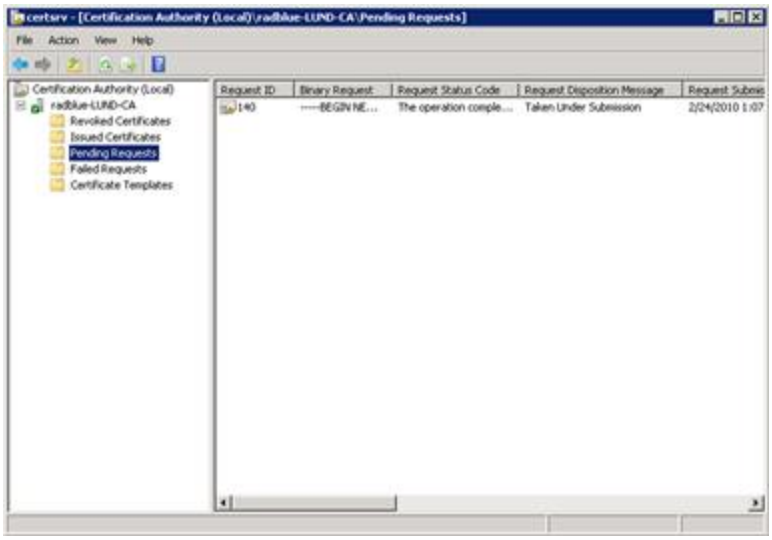
- **SCEP Server Polling Interval** - Type or select the interval, in milliseconds, in which the tool polls the certificate server until the tool's certificate request is approved.
- **Request SCEP Server Capabilities** - Select to request the options supported by the certificate authority server.
- **Request Certificate** - Click to request a certificate from the SCEP certificate authority server.

5. Click **Request Certificate**.

   The request certificate is sent to the SCEP server. The tool polls the SCEP server location defined in step 4 until a signed certificate is issued.

If the CA is using Microsoft Active Directory Certificate Services, follow these steps to issue a signed certificate on the CA:

a. From the computer where Microsoft Active Directory Certificate Services is installed, go to: **Start** > **Administrative Tools** > **Certification Authority**



b. Expand the server name, and click **Pending Requests**.

c. Click to highlight the certificate request.

d. Right-click the entry, and select **All Tasks** > **Issue**. When the certificate is issued, it disappears from the list.

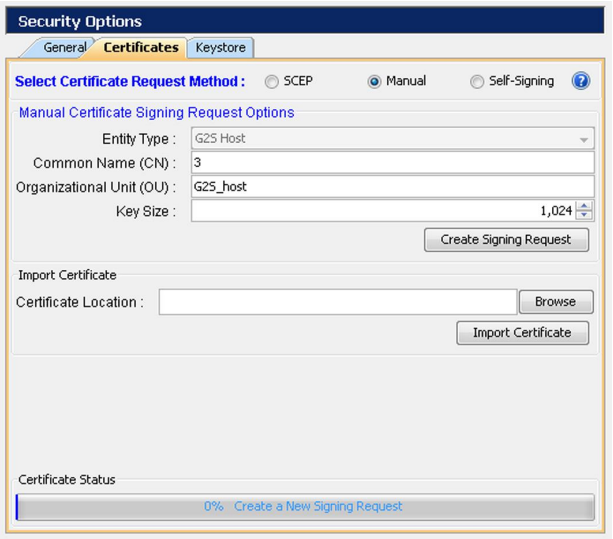
Once the certificate is signed, the RadBlue tool imports it the next time a poll is performed.

When the **Certificate Status** bar reads **100% Done!**, you have successfully imported the required certificate and can now use SSL messaging with the tool.

Load a Third-Party Certificate

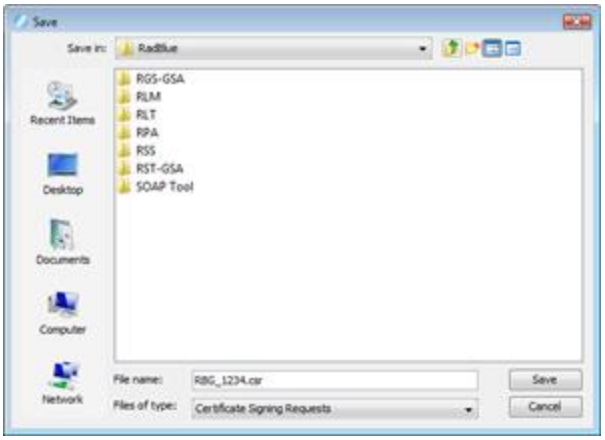
From the Certificates tab, you can create a certificate request that you can take to a certificate authority (CA) for signing. Once a signed certificate is available, you can then import it into the tool. Note that you must import both a certificate and CA certificate. They may come in one file or two, depending on the CA server. If you have two certificate files, you must import them one at a time.

1. From **Tools** > **Configure** > **Security Options**.
2. Click **Certificates**.
3. Select **Manual** as the **Certificate Request Method**.



5. Create a signing request by configuring the following fields with your request-specific information:
   - **Entity Type** - Click the drop-down arrow, and select the role of the tool: G2S Host, G2S EGM Proxy, G2S EGM, Other G2S.
   - **Common Name** - Type the tool's common name. In the case of an EGM, the common name would be the EGM identifier.
   - **Organizational Unit** - Type the organizational unit (role) of the tool: G2S_host, G2S_ egmProxy, G2S_egm, or Other G2S. By default, this field is populated with a value that corresponds to the Entity Type.
   - **Key Size** - Click the drop-down arrow, and select the size of the key pair supported by your network environment.

2. Click **Create Signing Request** to generate a signing request.



3. Navigate to the location where you want to save the certificate request file.
4. Modify the file name and file type as required.
5. Click **Save**.

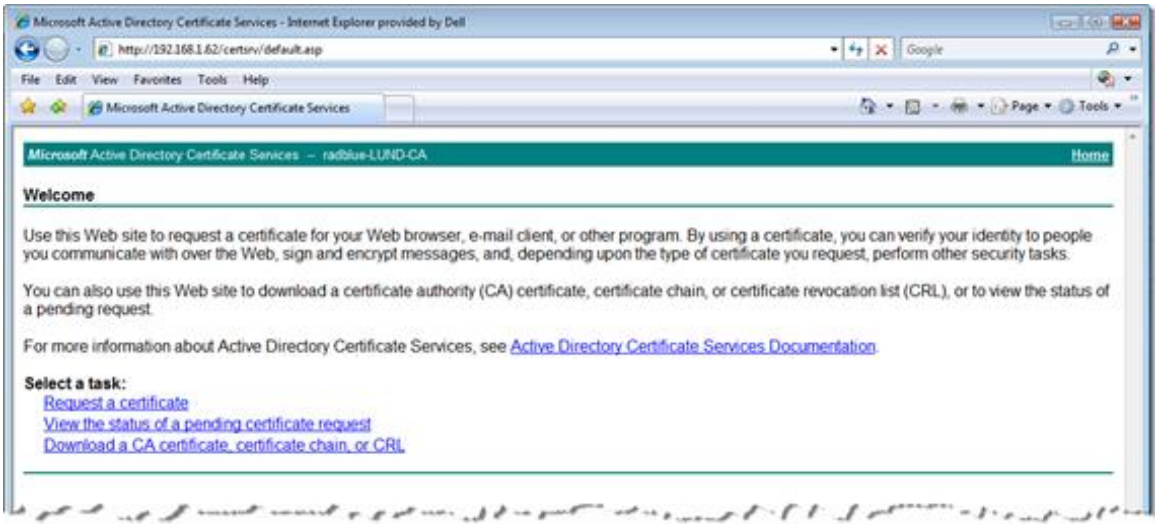
6. Click **OK**.
   Notice that the Certificate Status is updated.
7. Depending on the certificate authority you are using, you must now use the certificate request you created to obtain a signed certificate from a certificate authority. For an example of how to obtain a signed certificate from Microsoft Active Directory Certificate Services, see Obtaining a Signed Certificate Using Microsoft Active Directory Certificate Services.
8. Once you have a signed certificate that you can access, type the **Certificate Location** *or* click **Browse** to navigate to the signed certificate location.
9. Select the signed certificate file, and click **Open**.
10. Click **Import Certificate** to import the signed certificate.
11. If you have an additional certificate, repeat steps 8 through 10.
12. When the **Certificate Status** bar reads **100% Done!**, you have successfully imported the required certificate(s) and can now use SSL messaging with the tool.

Obtain a Signed Certificate Using Microsoft Active Directory Certificate Services

The following procedure is intended to provide an understanding of the process you may go through to create a signed certificate. Microsoft Active Directory Certificate Services is only one of many certificate authority programs. Your individual process may vary greatly depending on the certificate authority program you are using.

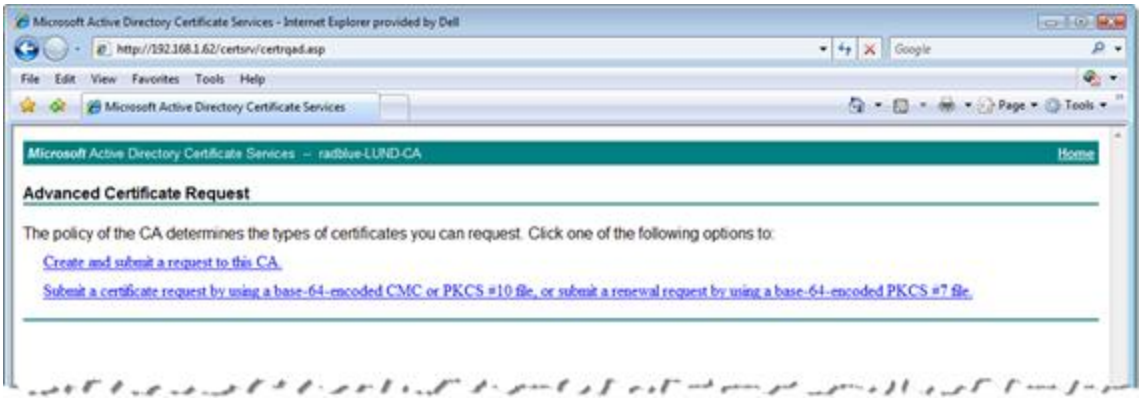
This procedure assumes that a certificate request has been created through the Third-Party tab on the Security Options screen.

1. Open the certificate request file in Notepad or Wordpad, and click anywhere inside the content.
2. Perform a **CTRL+a** to highlight all content and a **CTRL+c** to copy the content.
3. Open an Internet browser, enter the certificate authority location, and press **Enter**.

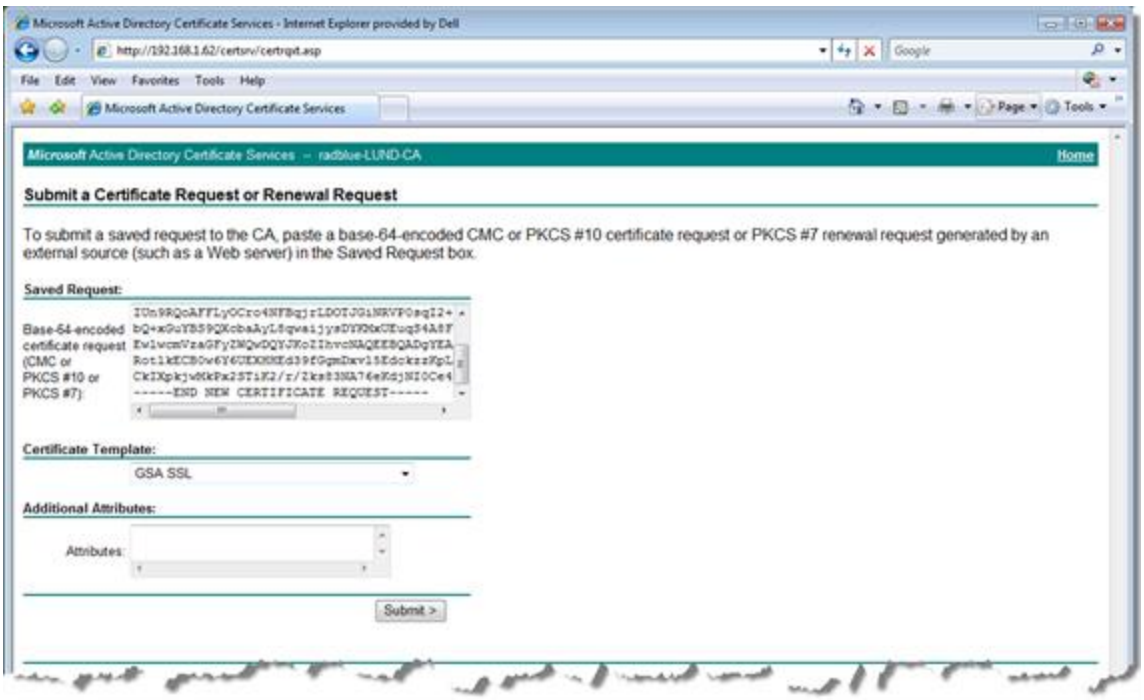


4. Click **Request a certificate**.

5. Click **advanced certificate request**.



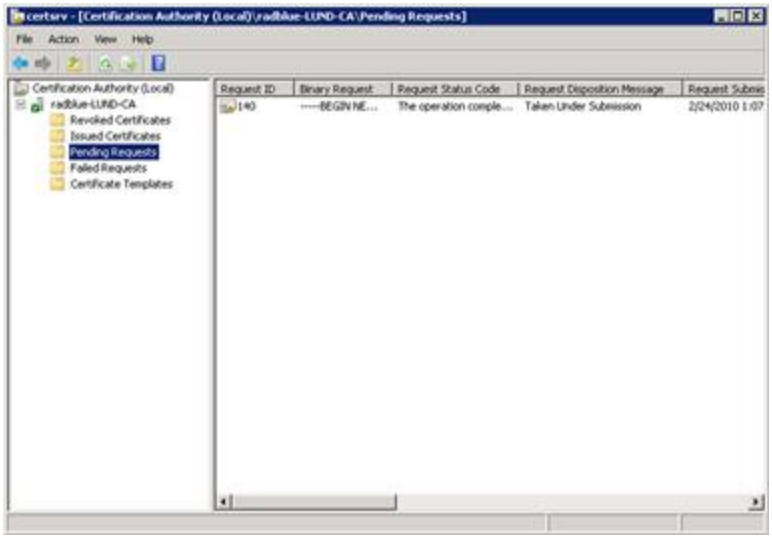
6. Click **Submit a certificate request by using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-54-encoded PKC #7 file**.



7. Click inside the **Base-64-encoded. . .** field and paste the certificate request content that you copied in step 2.
8. Click the **Certificate Template** drop-down arrow, and select the certificate template you use. In this example, we selected **GSA SSL**.

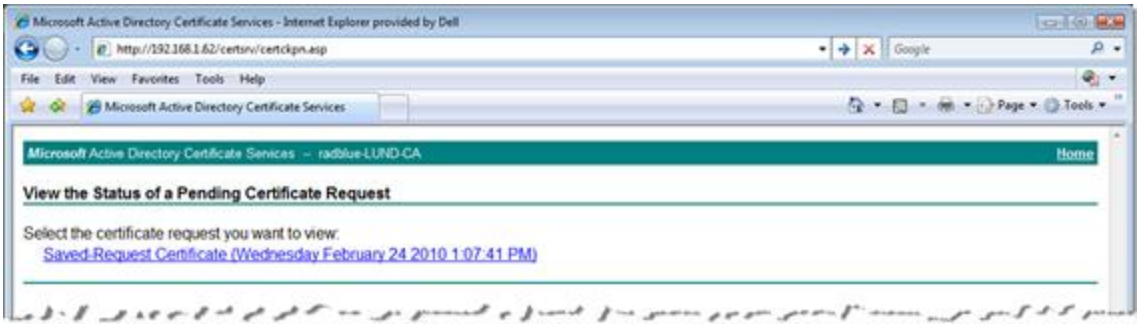
9. Click **Submit**.

10. Note the **Request ID**. In this case, the Request ID is **140**.
11. Minimize, but do not close, the browser.
12. From the computer where Microsoft Active Directory Certificate Services is installed, go to: **Start > Administrative Tools > Certification Authority**



13. Expand the server name, and click **Pending Requests**.
14. Click to highlight **Request ID 140**.
15. Right-click the entry, and select **All Tasks > Issue**. When the certificate is issued, it disappears from the list.
16. Maximize the browser window.
17. Click the **Home** link on the right-hand side of the page.

18. Click **View the status of a pending certificate request**.



19. Click **Saved-Request Certificate**.

20. Click **Download certificate chain** to download both parts of the signed certificate (certificate and CA certificate) as a single file. This is the recommended method because you only have one certificate to import into the tool.

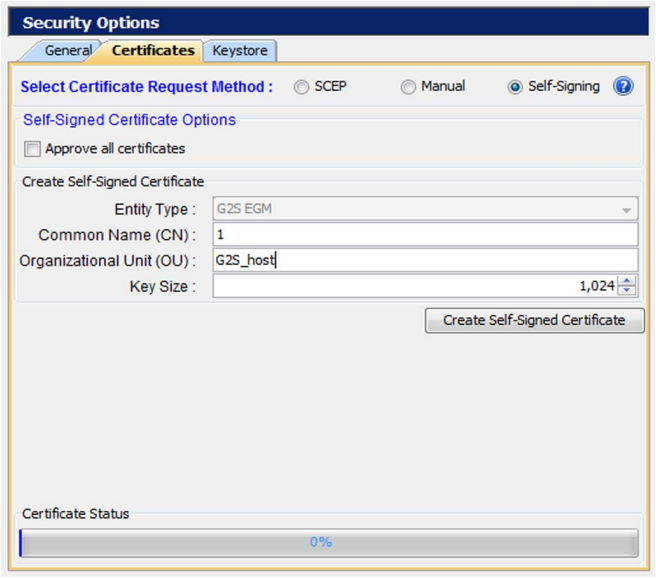
    If you want to create two separate files, click **Download certificate** to download the signed certificate file. Then, return to the home page and click **Download a CA certificate, certificate chain, or CRL**. Click **Download a CA certificate** to download the signed CA certificate file.

21. Once you have downloaded the certificate(s), open the tool and go to: **Configure** > **Security Options**
22. Select the **Third-Party** tab.
23. Click **Browse**, navigate to the signed certificate file, and click **Save**.
24. Click **Import Certificate** to import the selected certificate.
25. If you have an additional certificate, repeat steps 23 and 24.

Load a Self-Signing Certificate

The Certificates tab allows you to create a signed certificate for use with the tool.

1. From **Tools** > **Configure** > **Security Options**.
2. Click **Certificates**.
3. Select **Self-Signing** as the **Certificate Request Method**.



4. Select **Approve all certificates** to use SSL encryption without validating the certificate authority.

   If this option is cleared, the tool performs validity checking when an entity (for example, an EGM) initiates communications. The validity check includes:

   - Signed by *trusted* certificate authority?
   - Is current time/date within the period of validity (effective and expired date)?
   - Is issuer signature correct?
5. Configure certificate options as required.
   - **Entity Type** - Indicates the role of the tool: G2S Host (RGS), G2S EGM Proxy (RPA), G2S EGM (RST), Other G2S, or S2S Server (RSS). This information is determined by the tool. This field is *read-only*.
   - **Common Name** - Type the tool's common name. In the case of an EGM, the common name would be the EGM identifier. The tool will attempt to set this value.

- **Organizational Unit** - Type the organizational unit (role) of the tool: G2S_host, G2S_ egmProxy, G2S_egm, or Other G2S. By default, this field is populated with a value that corresponds to the Entity Type.
- **Key Size** - Click the drop-down arrow, and select the size of the key pair supported by your network environment.

6. Click to **Create Self-Signed Certificate** to generate a self-signed certificate based on the certificate options you completed.
7. When the **Certificate Status** bar reads **100% Done!**, you have successfully a signed certificate and can now use SSL messaging with the tool.

Manage Key Store Options

From the Key Store tab, you can select the type of key store file you want to use and manage installed key store files.

1. From the menu bar, click **Tools**, and select **Configure** to launch the Configuration screen.
2. Click **Security Options**.
3. Click the **Key Store** tab.
4. Click the **Select Key Store File** drop-down arrow, and select the type of key store file you want to use with the tool.

   Note: To update the available key store file types in the list, click **Refresh**.

6. To set the currently used certificate, click to highlight an installed certificate from the list and click **Set As Default**. The default value for this field is **<not set>**.
7. To remove an installed certificate, click to highlight the certificate, and click **Remove**.
8. To view the content of a certificate, click to highlight the certificate, and click **View**.

Import a PKCS #12 File

A PKCS #12 file is used to store multiple cryptography objects within a single file. The file commonly bundles a private key, with its X.509 certificate, or bundles all members of a chain of trust. The filename extension for PKCS #12 files is **P12** or **PFX**.

The **Import PKCS #12 File** option lets you quickly import the certificates stored in a P12 or PFX file into the tool's **client.jks** and **trusted.jks**. All certificates in the PKCS #12 file are imported to client.jks. Only non-key-entry certificates are imported to trusted.jks. Once the certificates are successfully imported, they can be viewed from the Key Store tab.

To import a PKCS #12 file:

1. Under the **Import PKCS #12 File** section on the **Key Store** tab, and click **Browse**.
2. Navigate to the **P12** or **PFX** file you want to import, and click **Open**.
3. Type the file password.
4. Click **Import**.

   The imported files are added to the list of key store files on the Key Store tab. Remember to use the **Select Key Store File** drop-down to switch between Trusted Key Store files and Client Key Store files.

Appendix B: Editing a SmartEGM Configuration File

About the SmartEGM Configuration File

The SmartEGM configuration file contains the RST data model. There are several default configurations that you can use and edit as needed. Each SmartEGM configuration file contains all available devices and attributes.

When you install RST, two separate instances are automatically created. There is a separate set of SmartEGM configuration files for each RST instance. Files for the second RST instance are appended with **-egm-2.xml** (for example, **smartegm-config-gsa-*egm-2*.xml**).

The available default configuration files include the following:

- **smartegm-config-gsa.xml** - Use this file to communicate with RGS. This is the master configuration file. All other configuration files are variations of one.
- **smartegm-config-gsa-central.xml** - Use this file for central game play. This file includes two central game play devices that correspond to game play devices on the EGM.
- **smartegm-config-gsa-ip.xml** - Use this file for informed player extensions.
- **smartegm-config-gsa-old.xml** - This file contains the previous (pre-19) SmartEGM configuration settings. Use this file if you do not want version 19+ changes and enhancements.
- **smartegm-config-gsa-rpa.xml** - Use this file to communicate with RPA.
- **smartegm-config-gsa-student-edition.xml** - For use with student licenses only. Use this file to communicate with RGS.
- **smartegm-config-gsa-student-edition-rpa.xml** - For use with student licenses only. Use this file to communicate with RPA.

Editing the SmartEGM Configuration File

You can edit the SmartEGM configuration file through an XML editor or by opening it as a text file. Edit the file as required. Be sure to save the file before closing.

1. Navigate to the SmartEGM configuration file:
   [**installation directory**] > **radblue** > **gsa** > **script** > **smart-conf** > **smart-egm**
2. Open the SmartEGM configuration file you want to use as a starting point in a text or XML editor.
3. Save the file with a new name, in the same directory.
4. Modify the configuration file as needed.
   - For information on editing the EGM identifier, see Editing the EGM ID.
   - For information on editing the host identifier list, see Adding or Modifying Hosts.
   - For information on adding game play devices, see Adding GamePlay Devices.
   - For information on modifying progressives, see Modifying Progressive Devices.
   - For information on adding unsupported events, see Adding Unsupported Events Elements.
   - For information on configuring offline ID validation, see Adding Offline Validation IDs.
   - For information on resetting the command ID sequence, see Resetting Command ID Sequence.
   - For information on adding game outcome, see Configure Game Outcome.
5. *Save and close the file*.
6. Launch RST.
7. Go to the **Configuration Control** section on the **Main** tab.
8. Click **Change SmartEGM Configuration**.
9. Click to select the configuration file you want to load, and click **Open**.
10. Click **Start SmartEGM** to start RST.

Editing the EGM ID

You can modify the SmartEGM EGM identifier through the SmartEGM configuration file.

Note that you can modify theSmartEGM EGM identifier through the RST user interface by clicking **Change SmartEGM ID** and entering a new EGM ID.

To change the EGM ID directly from the SmartEGM configuration file, replace the current **edm:egm-id** value with a valid G2S EGM ID.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<edm:config xmlns:edm="http://www.radblue.com/egm-data-model/schemas/v1.0.0/">
            <edm:egm edm:egm-id="RBG_1234" edm:g2s-schema-version="1.0.3"
                        edm:description="The Standard RadBlue SmartEGM">
```

Save and close the file before loading it into RST.

Adding or Modifying Hosts

You can modify the SmartEGM's registered host list through the SmartEGM configuration file.

The following table shows the default host list.

Host Index	Host ID	URL	Description	Registered?
0	0	localhost	Empty	False
1	1	http://localhost:31101/RGS/api-services/G2SAPI	RGS	True
2	0	localhost	Empty	False
3	0	localhost	Empty	False
4	0	localhost	Empty	False

You can change or define any of the existing four hosts by modifying the **edm:host-id**, **edm:url**, **edm:description** and **edm:host-registered** attributes. Each registered host ID must be unique and contain a valid URL (unless it is an empty row, signified by **host-id ="0"**).

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<edm:config xmlns:edm="http://www.radblue.com/egm-data-model/schemas/v1.0.0/">
<edm:egm edm:egm-id="RBG_1234" edm:g2s-schema-version="1.0.3"
            edm:description="The Standard RadBlue SmartEGM">
         <edm:host edm:host-id="1"
edm:url="http://localhost:31101/RGS/api-services/G2SAPI"
edm:description="The RGS" edm:host-registered="true" />
         <edm:host edm:host-id="0" edm:url="UNDEFINED"
edm:description="EMPTY" edm:host-registered="false" />
         <edm:host edm:host-id="0" edm:url="UNDEFINED"
edm:description="EMPTY" edm:host-registered="false" />
         <edm:host edm:host-id="0" edm:url="UNDEFINED"
edm:description="EMPTY" edm:host-registered="false" />
To add a new host, copy and paste an existing host entry. For example:
        <edm:host edm:host-id="0" edm:url="UNDEFINED"
                                  edm:description="EMPTY" edm:host-registered="false" />
```

Update the new host information as needed. Note that the *hostIndex* attribute, which is required, is handled automatically by RST, so you do not need to enter it when adding new host entries to the file.

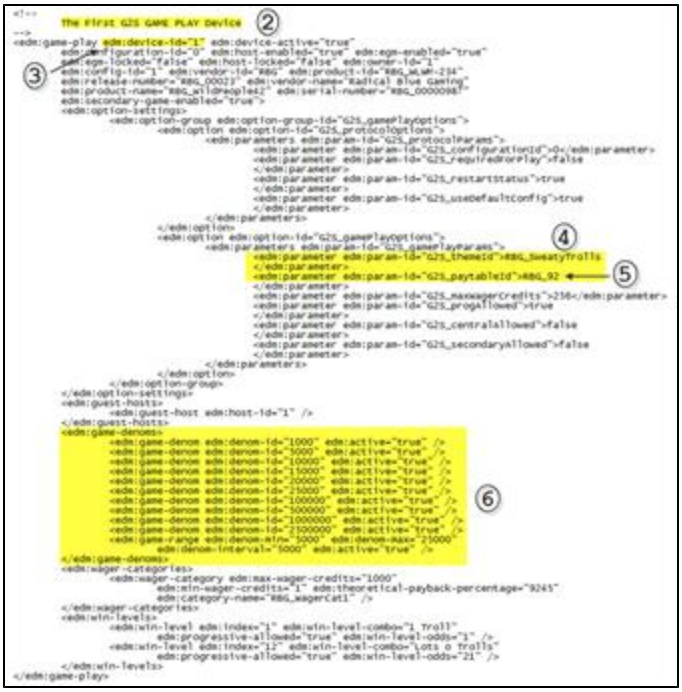
Save and close the file before loading it into RST.

Adding GamePlay Devices

You can add as many gamePlay devices as necessary to the SmartEGM configuration file.

To add a gamePlay device:

1. Copy and paste all of the content in an existing gamePlay device section.



2. Type a new name for the gamePlay device section.
3. Type a unique device identifier value for **edm:device-id**.
4. If required, change the game's theme by typing a new value for **edm:param-id="G2S_themeId"**.
5. If required, change the game's paytable ID by typing a new value for **edm:param-id="G2S_ paytableId"**.
6. Modify game denominations, found under the **edm:game-denoms** element, as needed.
7. Save and close the file.

Modifying Progressive Devices

You can modify progressive devices in the RST through the SmartEGM configuration file. For more information on the SmartEGM configuration file, see About the SmartEGM Configuration File and Editing the SmartEGM Configuration File.

A progressive device is owned by a progressive host, which identifies the progressive ID, a group of progressive levels or meters, and the levels in that group supported by the EGM. There is exactly one progressive ID for each progressive device, and one or more progressive levels for each progressive device.

Each progressive level can be hit by one or more win-level, in one or more gamePlay devices in the EGM. The win-level is identified by a win level index (a unique identifier that represents a combination of reel positions, for example, a full house or royal flush on a poker game).

The mapping between progressive level and win level in a game is:

progressiveId + progressiveLevel gamePlay device + winLevelIndex + wager (# credits * denom)

> **Note:** The `progressive.setProgressiveWin` command with a *payMethod* of **payHandpay** and **payVoucher** is not supported.
>
> The `bonus.setBonusAward` command with a *payMethod* of **payHandpay** and **payVoucher** is not supported.

Configuring Progressives in the SmartEGM

Navigate to:[**installation directory**] > **radblue** > **gsa** > **script** > **smart-conf** > **smart-egm**

Right-click **smartegm-config.xml**, and select **Edit**.

To configure progressives in RadBlue tools, you first need to configure the various win levels supported by the gamePlay device. These values are typically characteristics of the game, so they are not configurable through the G2S optionConfig class. The following excerpt is from gamePlay device 1 in the sample smartegm-config.xml file distributed with the tools:

```
gamePlay Device Section
        <edm:win-levels>
         <edm:win-level edm:index="1" edm:win-level-combo="1 Troll"
                                      edm:progressive-allowed="true" edm:win-level-odds="1"
/>
         <edm:win-level edm:index="12" edm:win-level-combo="Lots o Trolls"
                                      edm:progressive-allowed="true" edm:win-level-
odds="21" />
         </edm:win-levels>
```

Next, you need to configure the progressive device to tie a progressive level (meter) to a particular gamePlay device and win-level. This is configurable through G2S and can be done using the `setOptionList` command, or by editing the smartegm-config.xml file for the progressive device you want to hit when the appropriate winning combination is hit in the gamePlay device. A progressive data table is used in the progressive device to define the relationship of the progressive level and the gamePlay win level:

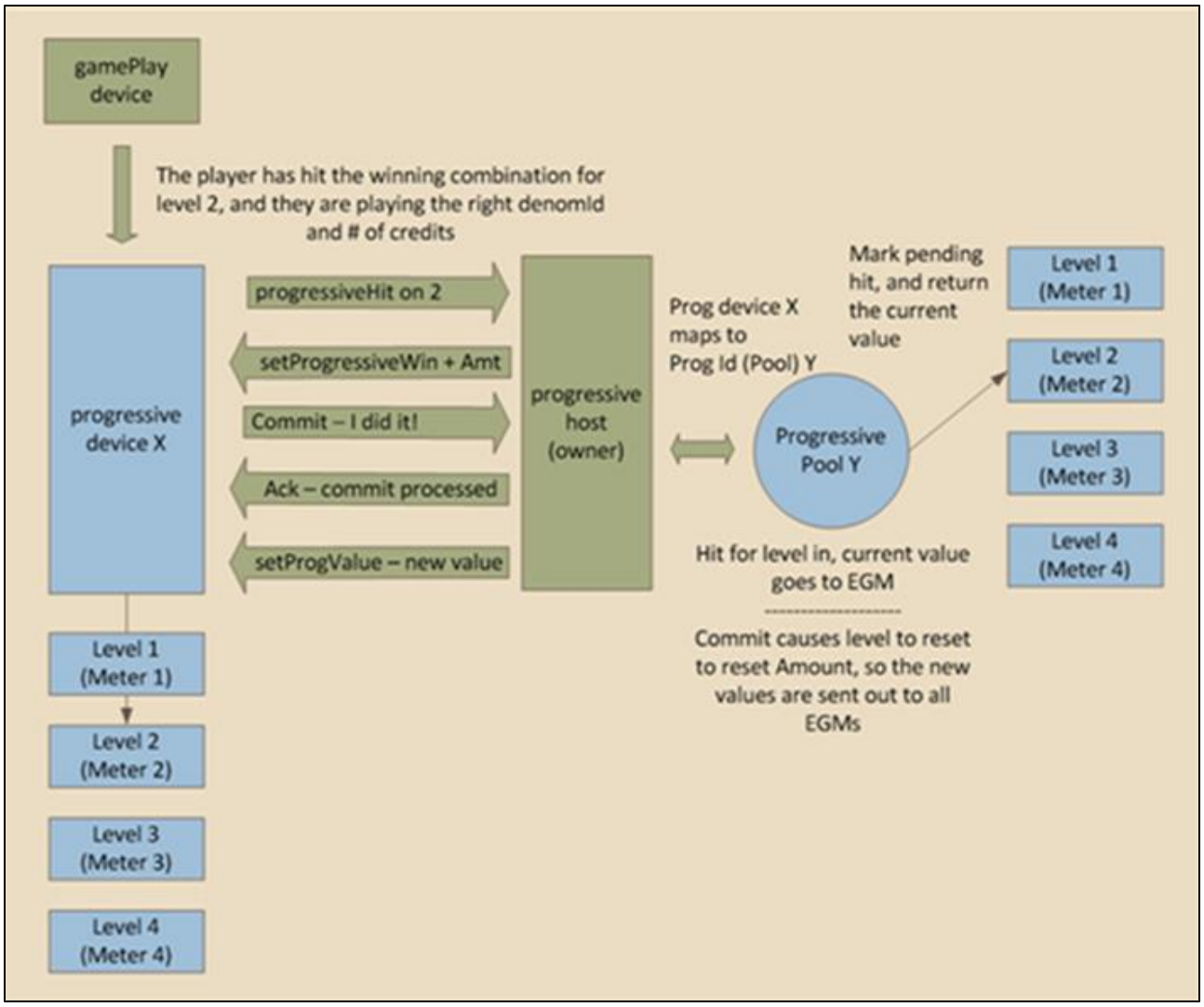
```
Progressive Device Section - Progressive Data Table
        <edm:option edm:option-id="G2S_progDataTable">
                <edm:parameters-table>
                        <edm:parameters edm:param-id="G2S_progData">
                        <edm:parameter edm:param-id="G2S_denomId">100000</edm:parameter>
                        <edm:parameter edm:param-id="G2S_themeId">
                                  RBG_sweatyTrolls</edm:parameter>
                        <edm:parameter edm:param-id="G2S_paytableId">RBG_92</edm:parameter>
                        <edm:parameter edm:param-id="G2S_numberOfCredits">3</edm:parameter>
                        <edm:parameter edm:param-id="G2S_levelId">1</edm:parameter>
                        <edm:parameter edm:param-id="G2S_gamePlayId">1</edm:parameter>
                        <edm:parameter edm:param-id="G2S_winLevelIndex">1</edm:parameter>
       </edm:parameters>
```

To modify the amount of time that the SmartEGM waits for a `setProgressiveValue` message from the progressive host (if at all), select the `G2S_noProgInfo` parameter and change the value to the amount of time, in milliseconds, that you want RST to wait for interval messages from the host. This time should be greater than the interval setting in the host. Zero (0) disables this feature.

```
Progressive Device Section - Option Settings
<edm:option-settings>
                  <edm:option-group edm:option-group-id="G2S_progressiveOptions" edm:option-
group-name="G2S                                   Progressive Options">
<edm:option edm:option-id="G2S_protocolOptions">
<edm:parameters edm:param-id="G2S_protocolParams">
<edm:parameter edm:param-id="G2S_timeToLive">0</edm:parameter>
<edm:parameter edm:param-id="G2S_requiredForPlay">false</edm:parameter>
<edm:parameter edm:param-id="G2S_configurationId">0</edm:parameter>
<edm:parameter edm:param-id="G2S_restartStatus">true</edm:parameter>
<edm:parameter edm:param-id="G2S_useDefaultConfig">true</edm:parameter>
<edm:parameter edm:param-id="G2S_progId">10</edm:parameter>
<edm:parameter edm:param-id="G2S_noResponseTimer">30000</edm:parameter>
<edm:parameter edm:param-id="G2S_noProgInfo">0</edm:parameter>
</edm:parameters>
</edm:option>
```

Sample Progressive Message Flow

The event that updates the host's progressive database is `G2S_PGE101` (Progressive Money Wagered). If this event is not generated by the EGM (not supported or not subscribed to by the host), the host's progressive values won't update when a game is played. The `setProgressiveValue` command is then used by the host to send the updated progressive values back to the EGM.



Unsupported Events Element Example

In this example, all gamePlay events are unsupported.

```
<edm:event-handler edm:device-id="1" edm:device-active="true" edm:configuration-id="0"
edm:host-enabled="true" edm:egm-enabled="true" edm:egm-locked="false" edm:host-
```

```
locked="false" edm:chatty="true" edm:owner-id="1" edm:config-id="1" edm:vendor-id="Unknown"
edm:product-id="Undefined" edm:release-number="Undefined" edm:vendor-name="Undefined"
edm:product-name="Undefined" edm:serial-number="Undefined">
        <edm:unsupported-events>
                <edm:unsupported-event edm:event-code="G2S_GPE" />
        </edm:unsupported-events>
</edm:event-handler>
```

Adding Unsupported Events Elements

The Unsupported Events element lets you specify which events are not supported by the SmartEGM for the host subscription. By default, the SmartEGM supports all events. The Unsupported Event element should be included in each eventHandler device that you want to affect.

Note that forced event subscriptions supersede unsupported events.

Name	Type	Use	Restrictions	Description
unsupported-events	element	optional	minOcc: 0 maxOcc: 1	Unsupported events element. This element can contain multiple unsupported-event elements. Each eventHandler device may contain only one unsupported-events element.
unsupported-event	element	required	minOcc: 1maxOcc:unbound	Single unsupported event element. This element can contain only one event-code attribute. However, depending the attribute value may specify multiple events, depending on how it is expressed.
event-code	attribute	required	xs:string	Event type or specific event to be excluded. Regular expressions are permissible.

Configure Offline ID Validation

You can configure offline ID validation support through the SmartEGM configuration file to test offline ID validation handling in the host system, specifically, `G2S_IDE105` (Unable to Validate ID Offline) and `G2S_IDE104` (ID Validated Offline).

To configure offline validation ID to the SmartEGM configuration file, go to the G2S ID Reader Device section of the file.

```
<edm:option edm:option-id="G2S_idReaderOptions">
<edm:parameters edm:param-id="G2S_idReaderParams">
```

```
                                                <edm:parameter edm:param-id="G2S_
idReaderTrack">1</edm:parameter>
                                                 <edm:parameter edm:param-id="G2S_
idReaderType">G2S_magCard
                                                 </edm:parameter>
                                                 <edm:parameter edm:param-id="G2S_
idValidMethod">G2S_host
                                                 </edm:parameter>
                                                 <edm:parameter edm:param-id="G2S_
msgDuration">15000
                                                 </edm:parameter>
                                                 <edm:parameter edm:param-id="G2S_
removalDelay">300000
</edm:parameter>
                                                <edm:parameter edm:param-id="G2S_
validTimeOut">300000
                                                 </edm:parameter>
<edm:parameter edm:param-id="G2S_waitTimeOut">15000
                                                 </edm:parameter>
                                                 <edm:parameter edm:param-id="G2S_
egmPhysicallyControls">true
                                                 </edm:parameter>
                                                <edm:parameter edm:param-id="G2S_
limitLosses">false
                                                </edm:parameter>
<edm:parameter edm:param-id="G2S_offLineValid">true
                                                </edm:parameter>
                        </edm:parameters>
</edm:option>
```

Modify the *G2S_waitTimeOut* and *G2S_offLineValid* attributes as needed. The *G2S_waitTimeOut* attribute indicates the amount of time RST waits for a response to the `getValidationId` command before it times out. The *G2S_offLineValid* attribute indicates whether RST allows offline ID validation.

Scroll down further in the ID Reader Device section, and modify `offline-patterns` element example or add new ID type patterns.

```
<edm:offline-patterns>
<edm:offline-pattern edm:id-type="G2S_player" edm:pattern="[0-9]{8}" />
</edm:offline-patterns>
```

Save and close the file before loading it into RST.

Resetting Command ID Sequence

The `edm:command-id-reset` attribute lets you reset the Command ID sequence to **1** just before the `commsOnline` command is sent. Set this attribute to **true** to reset the Command ID or set this **false** if you do not want to reset the Command ID. By default, this attribute is set to **false**.

In the following example, the command ID will reset before the `commsOnline` command is sent.

```
<!-- The G2S COMMUNICATIONS Device -->
<edm:communications edm:device-id="1" edm:device-active="true" edm:configuration-id="0"
edm:comms-on-line-ack-timeout="30000" edm:host-enabled="true" edm:egm-enabled="true"
edm:egm-locked="false" edm:host-locked="false" edm:owner-id="1" edm:config-id="0"
edm:vendor-id="Unknown" edm:product-id="Undefined" edm:release-number="Undefined"
edm:vendor-name="Undefined" edm:product-name="Undefined" edm:serial-number="Undefined"
edm:command-id-reset="true">
</edm:communications>
```

Configure Game Outcome in the SmartEGM Configuration File

Navigate to:[**installation directory**] > **sample** > **smart-egm**

Right-click **smartegm-config-gsa.xml**, and select **Edit**.

To configure game outcome in the SmartEGM configuration file, you must modify the wager-categories element, win-levels element and paytable element for each gamePlay device.

wager-categories Element

This element defines the wager category table for the gamePlay device. There is one wager-category element for each wagerCategoryItem as defined in the G2S specification.

wager-categories Elements

Element	Restrictions	Description
wager-category	minOcc: 1 maxOcc: ∞	Contains a wager category that is supported by the gamePlay device.

wager-category Element

This element defines a wager category by specifying the category name, wager credit limits and theoretical payback percentage.

wager-category Attributes

Attribute	Restrictions	Description
category-name	type: string use: required	Wager category identifier.
max-wager-credits	type: integer use: required	Maximum wager credits for the selected wager category.
min-wager-credits	type: integer use: required	Minimum wager credits for the selected wager category.
theoretical-payback-percentage	type: decimal use: required	Theoretical payback percentage of associated wager category.

win-levels Element

This element defines the win level table for the gamePlay device. There is one win-level element for each winLevelItem as defined in the G2S specification.

win-levels Attributes

Attribute	Restrictions	Description
win-level-matcher	type: string use: optional default: "Default"	Unique identifier that specifies the code in the SmartEGM that determines the outcome token. Currently only Default is defined.

win-levels Elements

Elements	Restrictions	Description
win-level	minOcc: 1 maxOcc: ∞	Contains a win level that is supported by the game play device.

win-level Element

This element defines the win level in the gamePlay device.

win-level Attributes

Attribute	Restrictions	Description
index	type: positive integer use: required	Unique paytable index for the win level.
win-level-combo	type: string use: optional default: " "	Description of paytable win level.
progressive-allowed	type: boolean use: optional default: "false"	Indicates whether the paytable win level can be assigned to a progressive game play device.
win-level-odds	type: positive integer use: optional default: "0"	Theoretical weight of associated win level.
win-level-token	type: string use: optional default: "Default"	Game outcome token that is associated with this win level.

paytable Element

This element defines the paytable for the gamePlay device.

paytable Elements

Element	Restrictions	Description
paytable-win-level	minOcc: 1 maxOcc: ∞	Paytable entry.

paytable-win-level Element

This element associates a wager multiplier for each win level index and wager category. There must be a paytable-wager-category for each wager category specified in the wager category table for this gamePlay device.

paytable-win-level Attributes

Attribute	Restrictions	Description
win-level-index	type: positive integer use: required	Unique identifier within a paytable for a specific prize level.

paytable-win-level Elements

Element	Restrictions	Description
pyatable-wager-category	minOcc: 1 maOcc: ∞	Associates win level index and wager category to a wager multiplier.

paytable-wager-category Element

This child element of the paytable-win-level element specifies the wager multipler for the win level and wager category.

paytable-wager-category Attributes

Element	Restrictions	Description
category-name	type: string use: required	Wager category name.
wager-multiplier	type: positive integer use: required	Number that is multiplied by the wager to determine the game's initial win.

Example SmartEGM Configuration - Game Outcome

```
<!-- The Second G2S GAME PLAY Device -->
```

The first thing you must do when adding game outcome to the SmartEGM configuration file is to define the game (`theme-type`). Valid theme types are IGT_poker, IGT_reels, IGT_keno and IGT_none.

```
<edm:game-play edm:device-id="2" edm:device-active="true" edm:configuration-id="0" edm:host-
enabled="true" edm:egm-enabled="true" edm:egm-locked="false" edm:host-locked="false"
edm:owner-id="1" edm:config-id="1" edm:vendor-id="RBG"
edm:product-id="RBG_SweatT-0012" edm:release-number="RBG_ST0099" edm:vendor-name="Radical
Blue Gaming" edm:product-name="RBG_SweatingMightily12" edm:serial-number="RBG_00000123"
edm:secondary-game-enabled="false" edm:theme-type="IGT_poker">
```

Next, you must define the wager categories associated with the game play device.

```
<edm:wager-categories>
              <edm:wager-category edm:min-wager-credits="1" edm:max-wager-credits="4"
edm:theoretical-payback-percentage="9473"
                           edm:category-name="RBG_WagerCat1" />
              <edm:wager-category edm:min-wager-credits="5" edm:max-wager-credits="1000"
edm:theoretical-payback-percentage="9610"
                           edm:category-name="RBG_WagerCat2" />
 </edm:wager-categories>
```

Next, you must define the win levels for the game play device. For each win-level, you specify the win-level-token that will be used to associate the win-level to the paytable.

```
<edm:win-levels edm:win-level-matcher="Default">
                          <edm:win-level edm:index="1" edm:win-level-combo="Royal Flush"
edm:progressive-allowed="true" edm:win-level
                                       odds="40390" edm:win-level-token="PokerRoyalFlush" />
                          <edm:win-level edm:index="2" edm:win-level-combo="Straight Flush"
edm:progressive-allowed="true" edm:win-level
                                       odds="9148" edm:win-level-token="PokerStraightFlush"
/>
                          <edm:win-level edm:index="3" edm:win-level-combo="Four of a Kind"
edm:progressive-allowed="false" edm:win-level
                                      odds="423" edm:win-level-token="PokerFourOfAKind" />
                           <edm:win-level edm:index="4" edm:win-level-combo="Full House"
edm:progressive-allowed="false" edm:win-level-odds="86"
                                      edm:win-level-token="PokerFullHouse" />
                           <edm:win-level edm:index="5" edm:win-level-combo="Flush"
edm:progressive-allowed="false" edm:win-level-odds="90"
                                      edm:win-level-token="PokerFlush" />
                           <edm:win-level edm:index="6" edm:win-level-combo="Straight"
edm:progressive-allowed="false" edm:win-level-odds="89"
                                      edm:win-level-token="PokerStraight" />
                           <edm:win-level edm:index="7" edm:win-level-combo="Three of a Kind"
edm:progressive-allowed="false" edm:win-level
                                      odds="13" edm:win-level-token="PokerThreeOfAKind" />
```

```
                        <edm:win-level edm:index="8" edm:win-level-combo="Two Pairs"
edm:progressive-allowed="false" edm:win-level-odds="7"
                                    edm:win-level-token="PokerTwoPairs" />
                        <edm:win-level edm:index="9" edm:win-level-combo="Jacks or Better"
edm:progressive-allowed="false" edm:win-level
                                     odds="4" edm:win-level-token="PokerJacksOrBetter" />
</edm:win-levels>
```

Finally, you must associate the wager multiplier to a wager category within a win level.

```
              <edm:paytable>
                        <edm:paytable-win-level edm:win-level-index="1">
                                  <edm:paytable-wager-category edm:category-name="RBG_
WagerCat1" edm:wager-multiplier="250" />
                                  <edm:paytable-wager-category edm:category-name="RBG_
WagerCat2" edm:wager-multiplier="800" />
                        </edm:paytable-win-level>
. . .
                            <edm:paytable-win-level edm:win-level-index="9">
                                    <edm:paytable-wager-category edm:category-name="RBG_
WagerCat1" edm:wager-multiplier="1" />
                                    <edm:paytable-wager-category edm:category-name="RBG_
WagerCat2" edm:wager-multiplier="1" />
                            </edm:paytable-win-level>
                 </edm:paytable>
</edm:game-play>
```

About the Debug Console

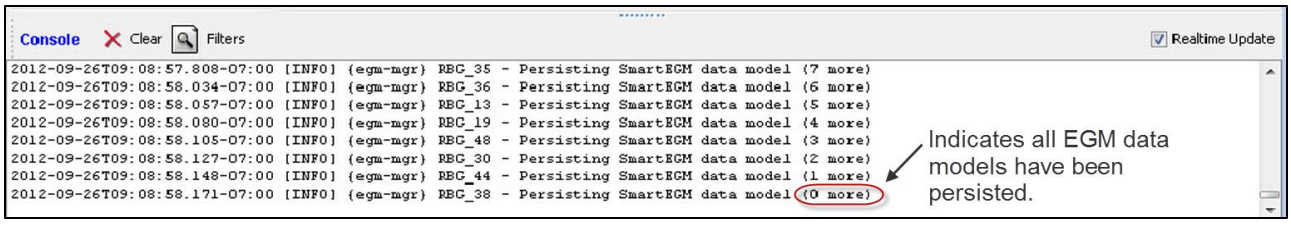
The Debug Console displays all informational, warning and critical errors that occur in the tool. Any of the following message types may appear in the Debug Console:

- **INFO** - Messages that do not impact the system, but may be useful to know. INFO messages appear in black type.
- **DEBUG** - Fine-grained informational events that are useful in troubleshooting. DEBUG messages appear in black.
- **ERROR** - Messages related to program errors. ERROR messages appear in red.
- **FATAL** - Designates a severe error events that will presumably lead the application to abort. FATAL messages appear in red.
- **UNKNOWN** - Messages that have not been assigned a logging designation. UNKNOWN messages appear in pink.
- **WARN** - Messages that indicate potentially harmful situations. WARN messages appear in blue.

You can clear the debug log and filter the debug log display (selectively display messages by warning level) as needed.

The information displayed in the Debug Console is written to a text file ([**tool name**].**txt**), located in the tool's logs directory.

Note that when you stop RLT, the Debug Console indicates when RLT has persisted (saved) each EGM's data model. When **(0 more)** is displayed, all EGMs have been persisted and you can exit RLT.



Clear the Debug Log Display

To clear the Debug Console display, click **Clear Log**.

Note that this option clears the display only, and not the text file associated with the Debug Log ([**tool name**]**.txt**, located in the tool's logs directory).

Filter Debug Messages

The Filter option lets you specify the type(s) of messages you want displayed in the Debug Console.

To filter Debug Console messages:

1. From the Debug Console, click **Filter**. The Logging Filter screen appears.
2. Click **Select All** to select all message types.

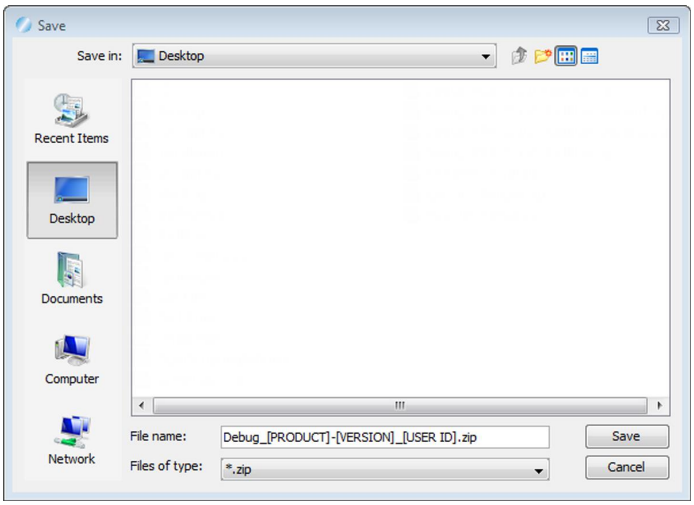
   or

1. Click **Clear All** to clear all boxes, and select the message types you want to display.
2. Click **Apply** for your changes to take effect, or click **Cancel** to exit the Logging Filter without applying any changes.

What to Do If You Can't Resolve an Error

The Export Debug option lets you create a ZIP file containing all the files that the RadBlue support team needs to troubleshoot product issues or for use in the RadBlue Analysis Suite (RAS). The ZIP file includes data from the time the tool was started to the time you select the Export Debug option.

1. Go to **File** > **Export Debug**.



2. A **Debug-[product-x.x.x]_[user ID].zip** file is exported to your computer's desktop.
3. Attach the ZIP file to an email, along with a description of the issue, and send it to support@radblue.com.

   or

   Go to www.radblue.com/support, complete the support form, attach the ZIP file and send.

   You will be contacted about your support issue within one business day.