radblue

RST User Guide

04 MAR 2014 - Version 38

About RadBlue

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.

About the RST Installation

RadBlue System Tester (RST) is available for both Windows and Linux operating systems.

For Windows, the installer installs two separate instances of RST to allow you to simulate two EGMs quickly and easily.

- The first instance is associated with the desktop icon, but can also be accessed through the RadBlue installation directory's **bin** sub-directory (**RST.exe**).
- The second instance can be launched from the **EGM-2.exe** file, also located in the **bin** sub-directory. Note that the second RST instance has its own set of SmartEGM configuration files.

Pre-Installation Requirements

Follow these guidelines before installing the RST.

1. If you are using a student license, not that the computer you install RST 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 RST license file on your computer prior to installing RST. If you are using a special version of RST, you must have a license for that version. If you have not received an RST license file, contact RadBlue Support.

If you are installing a version of the RST released prior to version 9 (April 7, 2010) on a Windows 2007 operating system, you may need to do additional configuration after completing the standard installation. See Bulletin 02: Configuring RST for Windows 2007

Computer Requirements

The minimum requirements for computers running the System Tester are:

- Operating System (32- or 64-bit): Windows (Vista or 7) or Linux
- Memory: 4 GB (minimum)
- Disk Space: 250 MB

Install RST on Windows

Follow these steps to install RST on a Windows operating system.

1. Double-click **RST_x_x_x.exe**.
2. Click **Next**.
3. Review the RadBlue agreement, and select **I accept the agreement** to consent to the agreement.
4. Click **Next**.
5. Type the location where you want the RST application installed, or click **Browse** to navigate to the location.
6. Click **Next**.

   Note: If you have a previous version of the tool installed, you are prompted to remove it before installing the new version. Click **Next** to uninstall the previous version before continuing with the new installation, or click **Back** to install the new version in a different directory.

7. Navigate to the location of the RST license file, and click **Next**.

   Note: For version 34 and higher, if you install a version of RST 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 RST over a previous installation. All components of the previous installation are removed by the installer except the license file and any backup files.

8. Select the **Start Menu** folder for RST.

   If you only want to create a shortcut for the current user, clear the Create shortcuts for all users checkbox.

   If you do not require a Start Menu folder for RGS, select Don’t create a Start Menu folder.

9. Click **Finish**.

Install RST on Linux

Follow these steps to install RST on a Linux operating system.

1. Download the tool's self-extracting install script from the RadBlue website onto your Linux computer.
2. Make the downloaded install script executable by typing the following command:
   chmod +x script_name.sh
3. Run the install script using the format:
   /script_path/script_name.sh
4. If you do not have Java installed on the computer, type **y** to install a JRE.
5. Click **Next**.
6. If you accept the RGS licensing terms, select **I accept the agreement**, and click **Next**.
7. Type the location of the directory where you want to install RGS, or click **Browse** to navigate to a location.
8. Click **Next**.

   Note: If you have a previous version of the tool installed, you are prompted to remove it before installing the new version. Click **Next** to uninstall the previous version before continuing with the new installation, or click **Back** to install the new version in a different directory.
9. Navigate to the location of the RGS license file, and click **Next**.

   Note: For version 34 and higher, if you install a version of RST 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 RST over a previous installation. All components of the previous installation are removed by the installer except the license file and any backup files.
10. Click **Finish**.

Uninstall RST

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

When RST is uninstalled, a backup folder is created in the RST directory that saves the installation's security and configuration parameters. When a subsequent RST 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.

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

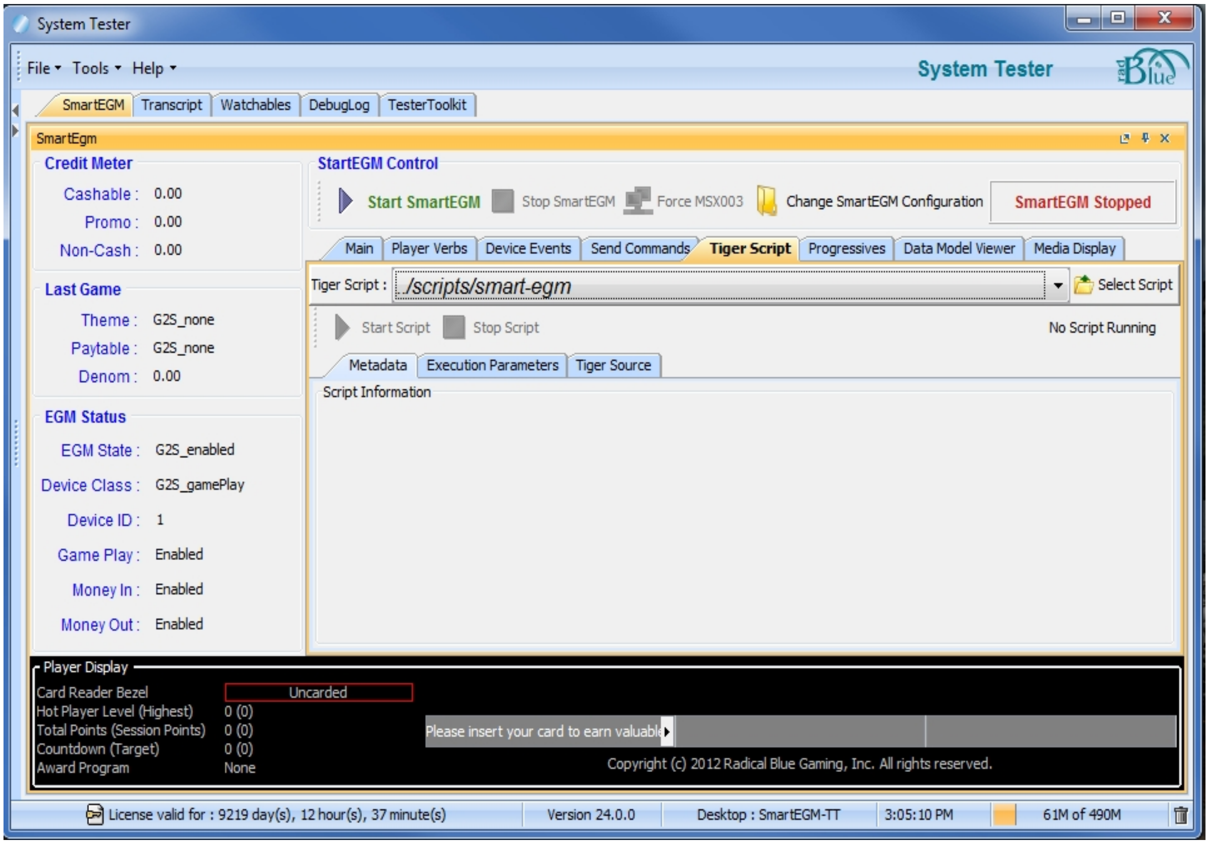
The backup folder is located in the RST installation directory. The following files are saved in the backup folder:

- All Java Keystore Files (JKS)
- EGM1 scep_config.xml
- EGM1 security_manager.xml
- EGM1 webserver.xml
- EGM1 Derby Port
- EGM1 SOAP IP
- EGM1 SOAP Port
- EGM1 SSL IP
- EGM1 SSL Port
- EGM2 scep_config.xml
- EGM2 security_manager.xml
- EGM2 webserver.xml
- EGM2 Derby Port
- EGM2 SOAP IP
- EGM2 SOAP Port
- EGM2 SSL IP
- EGM2 SSL Port

About RST

The RadBlue System Tester (RST) is the ultimate tool for G2S developers and testers. With the System Tester, you can simulate individual G2S tests as well as extended, customized testing. Using the various screen layouts and tabs, you can do any of the following plus more:

- Watch sent messages as they move from host to EGM in realtime and see the contents in an easy-to-read command format or in its original XML structure.
- Simulate human actions at the EGM using the Player Verbs tab.
- Run Tiger scripts (RadBlue ready made scripts) to automate various testing scenarios, or create your own scripts.
- Test and troubleshoot G2S messages.
- See all game play information.



Cashable : 0.00
Promo : 0.00
Non-Cash : 0.00

Theme : G2S_none
Paytable : G2S_none
Denom : 0.00

EGM State : G2S_enabled
Device Class : G2S_gamePlay
Device ID : 1
Game Play : Enabled
Money In : Enabled
Money Out : Enabled

SmartEGM Stopped

Tiger Script : ../scripts/smart-egm

No Script Running

Card Reader Bezel	Uncarded
Hot Player Level (Highest)	0 (0)
Total Points (Session Points)	0 (0)
Countdown (Target)	0 (0)
Award Program	None

Please insert your card to earn valuabl

Copyright (c) 2012 Radical Blue Gaming, Inc. All rights reserved.

License valid for : 9219 day(s), 12 hour(s), 37 minute(s) Version 24.0.0 Desktop : SmartEGM-TT 3:05:10 PM 61M of 490M

Additional Resources

- RST Release Notes
- RST, RPA and RGS Quick Start

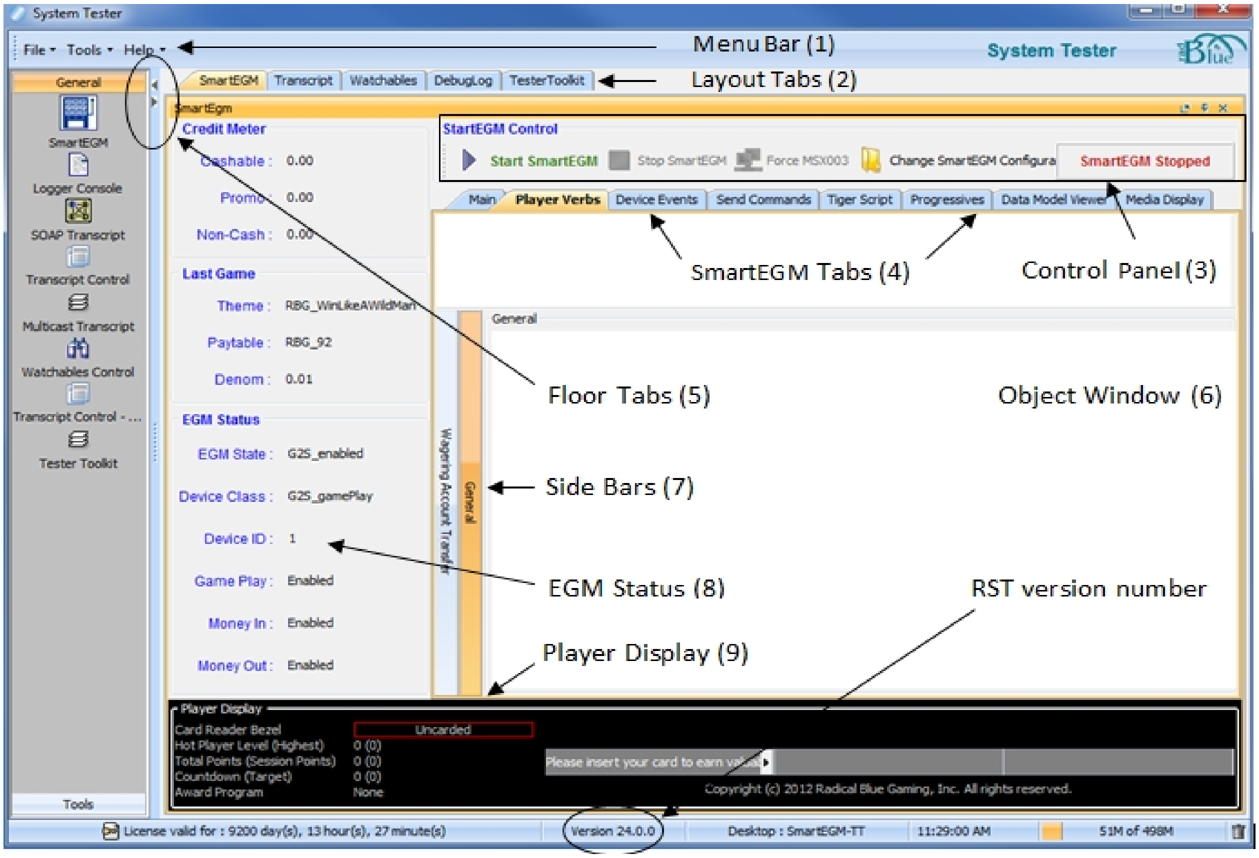
Supported GSA Versions

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

Protocol	Versions	
G2S	1.1.0	2.1.0

Review the RST Interface

Let’s look at the entire layout.



Credit Meter
Cashable : 0.00
Promo : 0.00
Non-Cash : 0.00

Last Game
Theme : RBG_WinLikeAWildMan
Paytable : RBG_92
Denom : 0.01

EGM Status
EGM State : G2S_enabled
Device Class : G2S_gamePlay
Device ID : 1
Game Play : Enabled
Money In : Enabled
Money Out : Enabled

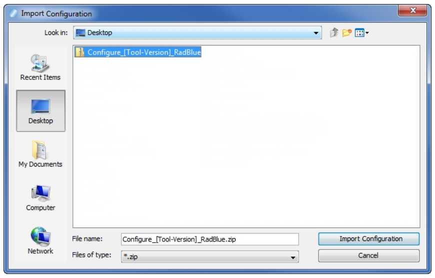
Player Display
Card Reader Bezel Uncarded
Hot Player Level (Highest) 0 (0)
Total Points (Session Points) 0 (0)
Countdown (Target) 0 (0)
Award Program None
Please insert your card to earn value
Copyright (c) 2012 Radical Blue Gaming, Inc. All rights reserved.

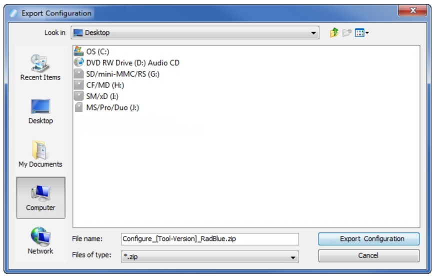
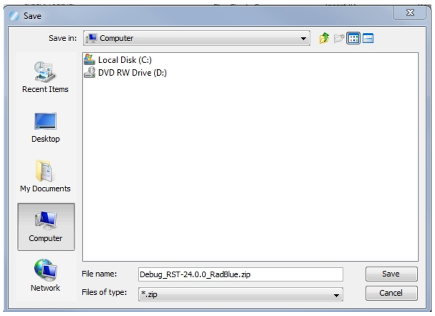
License valid for : 9200 day(s), 13 hour(s), 27 minute(s) | Version 24.0.0 | Desktop : SmartEGM-TT | 11:29:00 AM | 51M of 498M

Menu Bar (1)

From the menu bar you can access product options.

File

Menu Option	Description	Screen
Import Configure...	Select to import all configuration settings for the tool, exported from another version of the same tool, including security certificates. This option is used when you want to quickly set up a specific configuration for the tool that is already set up in another version of the tool.	 Import Configuration Look in: Desktop Configure_[Tool-Version]_RadBlue Recent Items Desktop My Documents Computer Network File name: Configure_[Tool-Version]_RadBlue.zip Files of type: *.zip Import Configuration Cancel

Menu Option	Description	Screen
Export Configure…	Select to export all configuration settings for the tool, including security certificates. The resulting ZIP file can then be imported into another version of the same tool.	 Look in: Desktop File name: Configure_[Tool-Version]_RadBlue.zip Files of type: *.zip Export Configuration Recent Items Desktop My Documents Computer Network OS (C:) DVD RW Drive (D:) Audio CD SD/mini-MMC/RS (G:) CF/MD (H:) SM/xD (I:) MS/Pro/Duo (J:) Export Configuration Cancel
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**.	 Save in: Computer File name: Debug_RST-24.0.0_RadBlue.zip Files of type: *.zip Save Recent Items Desktop My Documents Computer Network Local Disk (C:) DVD RW Drive (D:) Save Cancel
Exit	Select to close the product.	

Tools

Option	Short-cut	Description
Configure	F2	Select to see configuration options.
Toggle Floor Tab	F3	See *Floor Tabs* below.
GSA Message Validator	F4	Allows you to paste in a sample XML document and see if the message is valid against the selected schema.

Help

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

Layout Tabs (2)

Use these tabs to do the following:

Layout Tabs	Description
SmartEGM	Lets you simulate EGM activity and see the results. You can: • Send messages that simulate human activity or events that occur on EGM devices.

Layout Tabs	Description
	• Run scripts to automate various testing scenarios and repetitive testing tasks.
Transcript	Displays messages in real-time as they are sent and received in a command format or in its original XML structure. With SOAP and Multicast Transcripts, you see the SOAP wrapper and multicast messages.
Data Model Viewer	See simulated EGM messages as they are sent between the RST and host. Compare two data model snapshots and see changes that occur as you work through the testing process.
Watchables	Allows you to look for a specific attribute or value in that attribute. This feature is based on XML Path Language queries.
Debug Log	Shows all informational, warning, and critical errors that occur in the tool.
Tester Toolkit (module optional)	Extend normal testing by customizing messages sent by the RST. Test the SOAP layer as well as whether the host system can handle missing required values or illegal values.

Control Panel (3)

Use this panel to do the following:

Item	Description
Start SmartEGM	Begins sending RST messages. RST sends a `communications.getTransportOptions` command to the host.
Stop SmartEGM	Stops sending messages. RST sends a `communications.commsClosing` command to the host. The communications channel transitions from closing to closed.
Change SmartEGM Configuration	Use to change the G2S schema RST is currently using by looking at the G2S Schema field, located on the Main tab in the Configuration Control section. You can change the schema used by RST by modifying the SmartEGM configuration file. The latest version of the `smartegm-config.xml` file is loaded automatically upon start-up?
Force MSX003	Sends a `Communications Not Online` error to the host. Disconnects the communications channel between RST and the host. The message is marked as "lost," causing RST to respond with an MSX003 error to all messages from the host. Click **Stop SmartEGM** to clear the lost condition. In this case, the **Stop SmartEGM** does not send a `commsClosing` command.
SmartEGM Status box	A status box to the left of the SmartEGM Control options indicate whether the SmartEGM is running or stopped.

SmartEGM Tabs (4)

These tabs allow you to run scripts to automate EGM activity, or view EGM data.

- Main
- Player Verbs
- Device Events
- Progressives
- Send Commands
- Tiger Script (module optional)
- Media Display
- Data Model Viewer - Data Model Viewer displays the EGM's data model, allowing you to snapshot the current values and to compare two snapshots for any changes that may have occurred.

Floor Tabs (5)

Open/Close - Drag and drop the floor tab display objects onto the *Object Window (6)* when you want to create a custom desktop or layout.

Open this window using the arrows, or by going to **Tools** > **Toggle Floor Tabs**.

This panel contains views that you can see when you click the bars: General and Tools.

- When you drag and drop an object from one of these views, the object opens as a tab.
- Click and hold the tab to open it into a floating window.
- When you find a custom layout, save that desktop, go to **File** > **Save Desktop**

About Objects - Objects contain a single function (or group of functions) that you work with in the tool. They are populated from the tool's data model. The data model reflects all of the data that has been captured by the tool and any updates that are received while the tool is running.

When you first start the tool, all of the objects are empty. As messages are received by the tool, the appropriate objects are updated automatically. New objects are updated based on what's in the tool's data model.

As a result, objects are immediately populated when dragged onto a layout, as long as the tool has been running for a while and has received the applicable command. The same behavior holds true when switching between desktops. If the command is in the data model, the object is automatically populated.

Object Window (6)

The object window displays content associated with the selected SmartEGM tab.

Side Bar (7)

These bars are available in certain views, such as the Player Verbs tab. Click the bar to open it.

As the screen opens, the bar moves to the right of the screen. Click the bar to close it. Watch the bar move back to the left side of the screen.

- **General** - Displays the icons used to simulate human actions, such as inserting a coin or note into an EGM.
- **Smart Card** - This side bar is enabled *only* when a `smartCard` device is included in the currently loaded SmartEGM configuration file. From this side bar you can perform several smart card functions, including entering a PIN, transfering smart card funds and modifying a smart card.
- **Wagering Account Transfer (WAT)** - When enabled, it shows cashable, promo, or non-cashable information being transferred either to or from a player's account.

EGM Status (8)

This panel displays current EGM settings. As you use the tabs: Player Verbs, Device Events and WAT transfers to send messages, the EGM Status information is updated.

Player Display (9)

The Player Display shows messages sent from the host to the EGM, including welcome, award, and session messages; and card-out ("goodbye") messages.

The host can use either ASCII character set or substitution tokens, special characters that display as pre-defined information (for example, player name or EGM ID) at the EGM.

If the EGM is not carded on a player-required token, the actual token text displays. For example, if you use the player account number token (%a), you would see "%a" displayed instead of a player number. Substitution tokens can be used for any message type, and are described in "Appendix E" of the G2S Message Protocol document.

Player Display

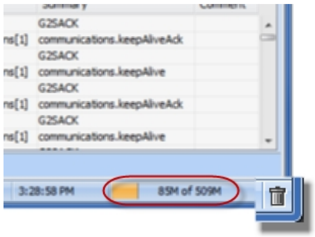
To assist testing efforts, player information is displayed in the left-hand corner of the screen.

Player Display	Description
Card Reader Bezel	Displays the inserted player card ID number. A green border indicates that a player card is inserted; a red border indicates that there is no player card inserted.
Hot Player Level	Displays the current hot player level. (Highest) - is the highest hot player level the player has obtained. This field resets at card-

Player Display	Description
	in and card-out, and does not require a player card-in for hot player determination to occur.
Total Points	Displays the total player bonus points (initial point balance + current session points). This field is initially populated by the `playerSessionStartAck` command. **Note**: if you send a `setPointBalance` command after a player session has started, the initial point balance is adjusted to the new value; the number sent is not added to the bonus point total. (Session Points) - indicates the bonus points accrued for the current session. The current session includes: base point awards, player point awards, generic override points awarded, and points awarded by the host. (The individual values are in the player log record). This field is only applicable to carded players.
Countdown	Used to show a carded player the progress toward the next point award. If the player stops playing before the countdown is reached, the countdown may be carried over to another play session, if configured. (Target) - the value the player must achieve to earn the award.
Award Program	At the end of a game session, the EGM calculates points- lowest cost per point (increment x target / award). In the screen example, NONE indicates there is no player card inserted, therefore, no points.

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.



	Summary	Comment
	G2SACK	
ns[1]	communications.keepAliveAck	
	G2SACK	
ns[1]	communications.keepAlive	
	G2SACK	
ns[1]	communications.keepAliveAck	
	G2SACK	
ns[1]	communications.keepAlive	

3:28:58 PM 85M of 509M

About the Main Tab

From the Main tab, you can load a SmartEGM configuration file and do the following:

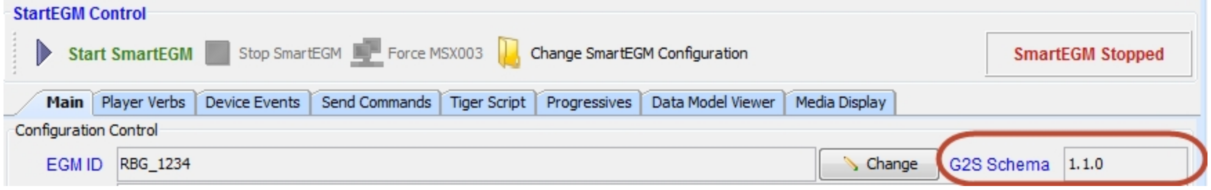
- Identify the G2S Schema you are currently using.
- Modify the EGM identifier, see Editing the EGM ID.

- Display a list of hosts and devices.
- Edit the Host ID, description, and URL, see Adding or Modifying Hosts.

Find the Currently Loaded G2S Schema

The schema version is located at: **Main** Layout tab > **Configuration Control** section.

You can change the schema used by RST by modifying the SmartEGM configuration file.



EGM ID RBG_1234

G2S Schema 1.1.0

Start Communications with Host Overview

Follow these steps to begin communicating with the host:

1. Connect RST to a Host.
   Edit Host ID and URL, to connect RST to a host in your lab.
   If you are using the RadBlue Student Edition, connect to RGS.
2. Load a SmartEGM configuration file.
   These default configuration files include:
   - EGM attributes
   - Host information
   - Schema version
3. Start the RST. Click **Start SmartEGM** on the Control Panel. For help, see RST Interface.

   Previously, when creating a new host through the `setCommConfig` command, RST created all five default host devices. RST automatically creates the devices (communications, eventHandler) and also includes (meters, optionConfig and GAT) because the RST supports them.
4. Send a G2S Command to the host.
   At this point, you can send any G2S command using these SmartEGM tabs:
   - Player Verbs
   - Device Events
   - Send Commands
   - Progressives - or-
   - Run automated scripts from the Tiger Script tab

   In this example, let's just send a simple command.
   - Click the **Send Commands** tab.
   - Click Send Option List icon.
5. Click the **Transcript** Layout tab to see the messages sent and received by RST. Examples:
   - Are the correct commands being sent?
   - Is the `commsOnline` command being sent during startup?
   - Are messages being acknowledged (with a G2SACK as well as a message ACK)?
   - Are there correct request-response pairs?

2. Stop the RST.
   Click **Stop SmartEGM**on the Control Panel. For help, see RST Interface.

   Note: You can now customize the RST.

Load a New SmartEGM Configuration File

Use the Change SmartEGM Configuration option to load a configuration file into the SmartEGM.

1. From the SmartEGM Main tab, click **Change SmartEGM Configuration**.
2. Click the configuration file you want to use, or select your own custom configuration file.
3. Click **Open**.

Save Your Configuration

If you want to save the configuration and reuse it the next time you restart RST, do the following:

1. Go to **File** menubar > **Save Desktop**.
2. Click **OK** to save the configuration.

Customizing the RST

Once you start testing, you may want to customize the RST to better suit your test objectives. You can customize in two ways: using the RST Interface or .xml file.

Using the RST Interface

Go to the menu bar, **Tools > Configure > Scripting Configuration** screen.

Some examples:

1. To add SSL security, go to the **Security** functional area.
2. To enable GZIP, go to **Engine Options**.

Functional Area	Description
Desktop	Modify Transcript, log, and file data
Engine	Change: • transport - IP, SOAP, URL, SSL Port and enable GZIP • meter rollover values • filters - selected information displayed in the Transcript window
Security	Enable and configure SSL encryption options
Email	Configure RST to send email alerts
License	View or load a new license

Using xml

Go to [**installation directory**] > **bin> RST.vmoptions file**

Functional Area	Description
Changing cabinet.setDateTime	By default, RST honors the `cabinet.setDateTime` command. This change affects all time values in RST except the date/time used in the Debug Log, which uses the CPU date/time. Change **Dcom.radblue.g2s.egm.cabinet.honorSetDateTime** line to **false**: `Dcom.radblue.g2s.egm.cabinet.honorSetDateTime=false`

About Player Verbs

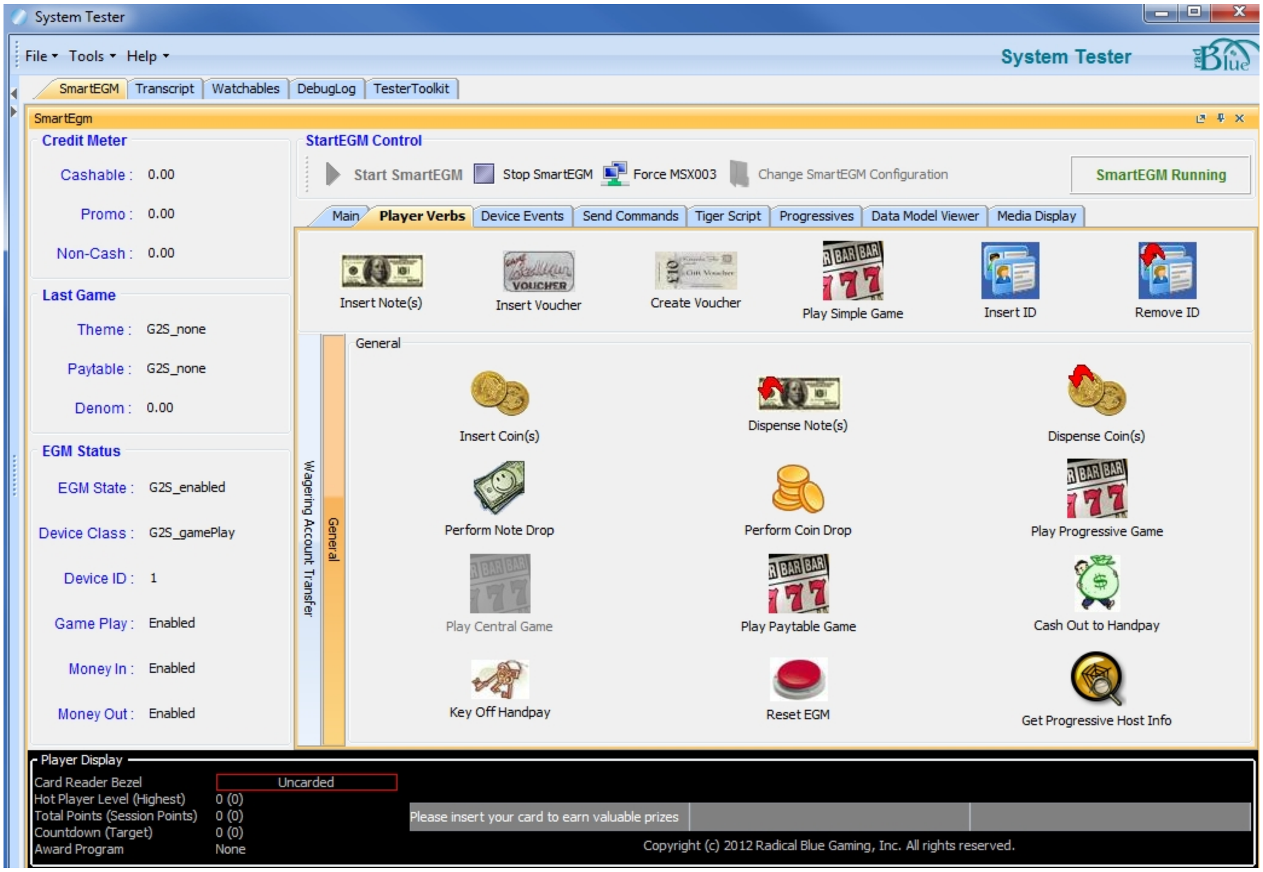
Use the Player Verbs to simulate EGM activity. Player Verbs allow you to:

- Simulate human activity (player and employee)at an EGM—to include WAT transactions
- Send messages that simulate events that occur on an EGM.

Each Player Verb button push represents an action at an EGM and may generate multiple messages to the host (RGS or your lab computer).

Always review the Transcript after you send each Player Verb, so you can see the flow of messages, and look at the EGM Status (left-side of the screen) for updates.

Verb options are dependent on licensing.



Credit Meter

Cashable : 0.00

Promo : 0.00

Non-Cash : 0.00

Last Game

Theme : G2S_none

Paytable : G2S_none

Denom : 0.00

EGM Status

EGM State : G2S_enabled

Device Class : G2S_gamePlay

Device ID : 1

Game Play : Enabled

Money In : Enabled

Money Out : Enabled

Player Display

Card Reader Bezel Uncarded

Hot Player Level (Highest) 0 (0)

Total Points (Session Points) 0 (0)

Countdown (Target) 0 (0)

Award Program None

Please insert your card to earn valuable prizes

Copyright (c) 2012 Radical Blue Gaming, Inc. All rights reserved.

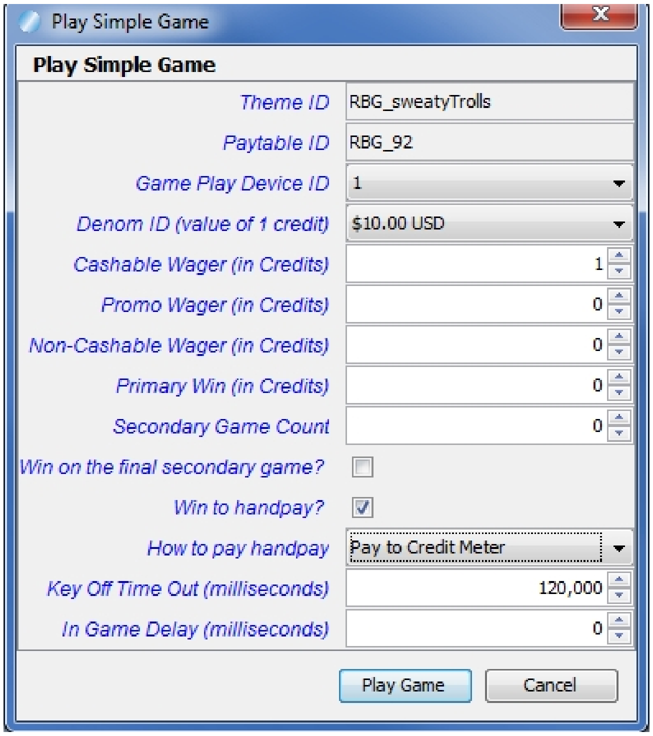
Using Player Verbs Overview

1. Click **Change SmartEGM Configuration** to change your configuration.
   For example, if you want to **Play a Central Game**, select **smartegm-config-gsa-central** configuration. When you change configuration, you must **Restart** the RST.
2. Click **Start SmartEGM.** to initiate communication with the host and enable Player Verb options.
3. Go to **SmartEGM layout > Player Verbs tab**.
4. Add money to the **Credit Meter** (top left of screen). The quickest way to add money to a meter is to click **Insert Notes(s)**.
   The Credit Meter must show a value for the following options: Play a Simple/Central/Progressive Game, Dispense Coins, Perform Note or Coin Drop, Cash Out to Handpay, or WAT.
5. Play a game. (May be optional.) Depending on your tests, choose to play a Simple, Central, or a Progressive game.
6. Click the **Player Verb** you want to simulate at the EGM.
7. Click the **Transcript** tab to see the message flow.

Play Simple Game

Simulate playing a non-progressive game at an EGM.

1. From the Player Verbs tab, click **Play Simple Game** and configure the Play Simple Game options as needed.
   - **Theme ID** - Read-only field.
     Shows current EGM theme. Dependent on the Game Play Device ID value.
   - **Paytable ID** - Read-only field.
     Shows current EGM paytable identifier. Dependent on the Game Play Device ID value.
   - **Game Play Device ID** - Select the game play device on which you want to play.
   - **Denom ID (value of 1 credit)** - Select the value of a single credit.
   - **Cashable Wager (in Credits)** - Type or select the number of cashable, non-promotional credits to wager.
   - **Non-Cashable Wager (in Credits)** - Type or select the number of non-cashable, non-promotional credits to wager.
   - **Promo Wager (in Credits)** - Type or select the number of promotional credits you want to wager.



Play Simple Game

Theme ID	RBG_sweatyTrolls
Paytable ID	RBG_92
Game Play Device ID	1
Denom ID (value of 1 credit)	$10.00 USD
Cashable Wager (in Credits)	1
Promo Wager (in Credits)	0
Non-Cashable Wager (in Credits)	0
Primary Win (in Credits)	0
Secondary Game Count	0
Win on the final secondary game?	☐
Win to handpay?	☑
How to pay handpay	Pay to Credit Meter
Key Off Time Out (milliseconds)	120,000
In Game Delay (milliseconds)	0

- **Primary Win (in Credits)** - Number of credits won after playing — 0 credits are permissible.
- **Secondary Game Count** - Type or select the number of secondary games played.
- **Win on the final secondary game?** If selected, all winnings from all secondary games are retained. If cleared, no secondary game winnings are retained.
- **Win to handpay?** - Select the win amount to result in a handpay condition.
- **How to pay handpay** - Select the type of handpay that occur after game play has concluded.

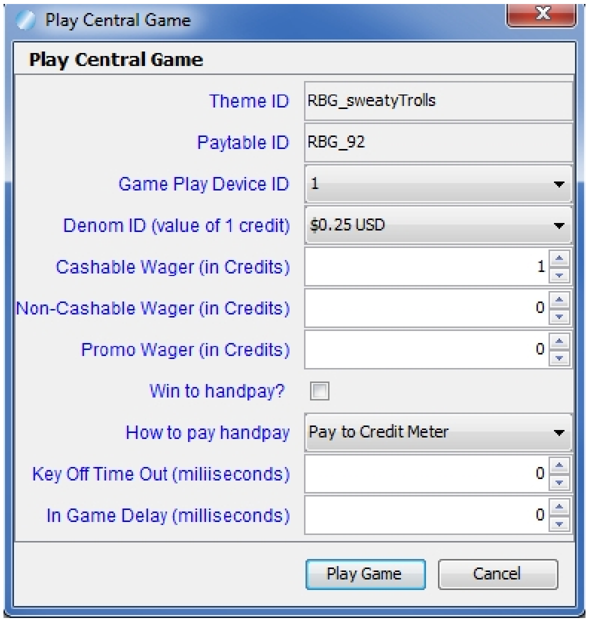
  If you select *Cancel a Handpay* or *Wait for Remote Key*, the SmartEGM is locked and the winning credits are keyed off (either through the Key Off Handpay verb or by sending a `setRemoteKeyOff` command to RST).
- **Key Off Time Out (milliseconds)** - Type or select the number of milliseconds before the EGM times out—waiting to receive a `setRemoteKeyOff` command.
- **In Game Delay (milliseconds)** - Type or select the number of milliseconds to delay when a game completes.

3. Click **Play Game**. Look at the Credit Meter decrement each time you click **Play Game**.

Play Central Game

Simulate game play at an EGM connected to a central server. Follow these steps to play a game to generate credit values.

1. Select central configuration to activate. Click **Change SmartEGM Configuration**.
2. From the Player Verbs tab, click **Play Central Game**
3. Configure the Play Game as needed — Lists for Game Play Devices and Denom ID lists, are located in the **centralGamePlay table** in the central device's profile.
   - **Theme ID** - Read-only field.
     Shows current EGM theme. Dependent on the Game Play Device ID value.
   - **Paytable ID** - Read-only field.
     Shows current EGM paytable identifier. Dependent on the Game Play Device ID value.
   - **Game Play Device ID** - Select the game play device on which you want to play.
   - **Denom ID (value of 1 credit)** - Select the value of a single credit.
   - **Cashable Wager (in Credits)** - Type or select the number of cashable, non-promotional credits to wager.
   - **Non-Cashable Wager (in Credits)** - Type or select the number of non-cashable, non-promotional credits to wager.
   - **Promo Wager (in Credits)** - Type or select the number of promotional credits you want to wager.



Play Central Game

Play Central Game

Theme ID	RBG_sweatyTrolls
Paytable ID	RBG_92
Game Play Device ID	1
Denom ID (value of 1 credit)	$0.25 USD
Cashable Wager (in Credits)	1
Non-Cashable Wager (in Credits)	0
Promo Wager (in Credits)	0
Win to handpay?	☐
How to pay handpay	Pay to Credit Meter
Key Off Time Out (miliiseconds)	0
In Game Delay (milliseconds)	0

- **Win to handpay?** - Select the win amount to result in a handpay condition.
- **How to pay handpay** - Select the type of handpay that occur after game play has concluded.

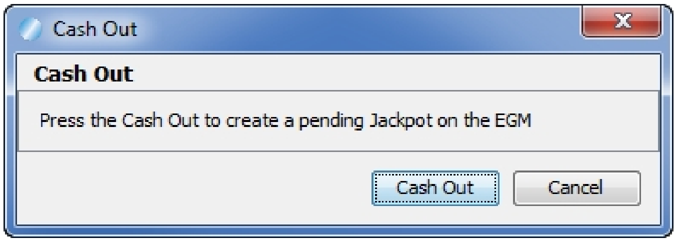
  If you select *Cancel a Handpay* or *Wait for Remote Key*, the SmartEGM is locked and the winning credits are keyed off (either through the Key Off Handpay verb or by sending a `setRemoteKeyOff` command to RST).
- **Key Off Time Out (milliseconds)** - Type or select the number of milliseconds before the EGM times out—waiting to receive a `setRemoteKeyOff` command.
- **In Game Delay (milliseconds)** - Type or select the number of milliseconds to delay when a game completes.

3. Click **Play Game**. Look at the Credit Meter decrement each time you click **Play Game**.

Cash Out to Handpay

Simulate a cash out handpay condition.

1. Add money to the Credit Meter.
   If the credit meter is zero (0), you cannot cash out to a handpay — instead you see the Cash Out Error: *You cannot cash out to handpay with no credits on the credit meters*.
2. (Optional.) Play a game.
3. From the Player Verbs tab, click **Cash Out to Handpay**.

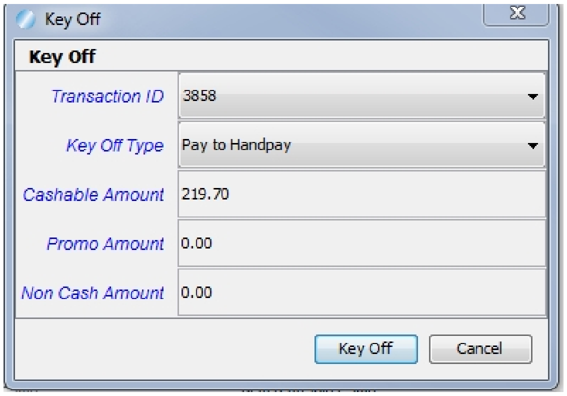


Cash Out

Cash Out

Press the Cash Out to create a pending Jackpot on the EGM

4. Click **Cash Out**.
5. Look at the **EGM Status** panel.
   - Cabinet EGM State changes to **G2S_egmLocked**
   - Device Class changes to **G2S_handpay**
6. Click **Key Off Handpay** to enable play on the EGM.
7. In the Key Off Type, ensure you select **Pay to Handpay**, then click **Key Off**.
8. Look at the **EGM Status** panel and notice the following:
   - The EGM State and Device Class are enabled.
   - The Credit Meter Cashable, Promo, or Non-Cash meters are 0.



Key Off

Key Off

Transaction ID: 3858

Key Off Type: Pay to Handpay

Cashable Amount: 219.70

Promo Amount: 0.00

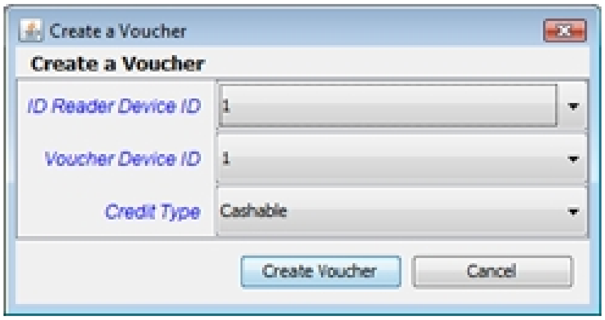
Non Cash Amount: 0.00

Create Voucher

Simulate a new voucher being created at an EGM. The voucher uses a `voucher-rst.xml` database, which are voucher numbers used to validate redeemed vouchers.

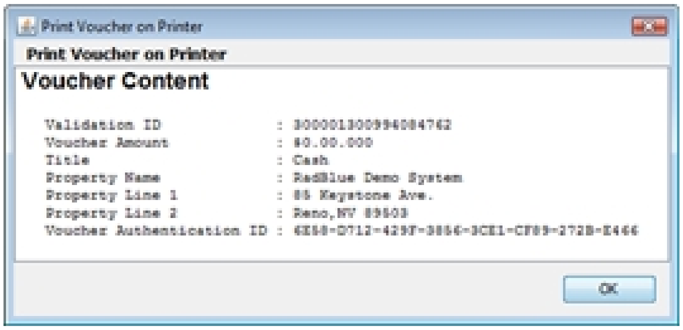
The voucher database is separate from the databases used with the new Tiger verbs.

1. Add money to the Credit Meter. You cannot create vouchers for zero (0) dollars. See Insert Note.
2. Ensure the EGM Status: EGM State: is enabled. (See the SmartEgm panel, below Credit Meter. See Reset EGM to unlock.)
3. From the Player Verbs tab, click **Create Voucher**.
   The `voucher-rst`.xml file is created automatically the first time you create a voucher. Each time you create a voucher, another voucher is inserted into the database.
2. Configure the Create a Voucher options as needed.
   - **ID Reader Device ID** - Select the associated device identifier.
   - **Voucher Device ID** - Select the identifier of the voucher device.
   - **Credit Type** - Select whether the voucher is cashable, non-cashable or promotional.



ID Reader Device ID: 1
Voucher Device ID: 1
Credit Type: Cashable

3. Click **Create Voucher**. You see the voucher that normally is printed.



Voucher Content

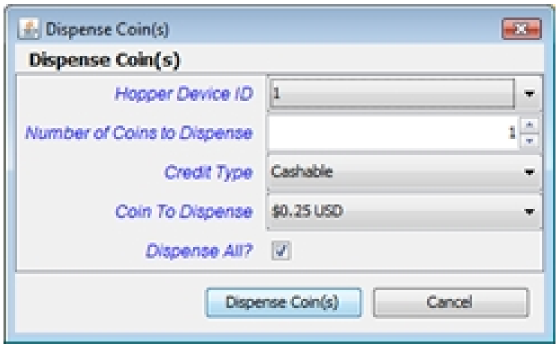
```
Validation ID               : 300001300994084762
Voucher Amount              : $0.00.000
Title                       : Cash
Property Name               : RedBlue Demo System
Property Line 1             : 85 Keystone Ave.
Property Line 2             : Reno,NV 89503
Voucher Authentication ID : 4E58-D712-429F-3856-3CE1-CF89-272B-E466
```

4. Click **OK**. You are now ready to the Insert Voucher.

Dispense Coin(s)

Simulate the dispensing of coins or tokens from an EGM.

1. Add $10.00 to the Cashable Credit Meter. See Insert Notes.
2. From the Player Verbs tab, click **Dispense Coin(s)**.
3. Configure the Dispense Coin(s) options as needed.
   - **Hopper Device ID** - Select the identifier of the hopper device.
   - **Number of Coins to Dispense** - Type or select the number of coins to be dispensed.
   - **Credit Type** - Select whether the credits dispensed are cashable, non-cashable or promotional.
   - **Coin To Dispense** - Select the denomination of the coins to be dispensed.
     Coin denoms are defined in the G2S Coin Acceptor Device section of the `smart-egm.xml` file.
   - **Dispense All?** - If checked, all credits are dispensed, regardless of classification. This field overrides values set in the following fields:
     - Number of Coins to Dispense
     - Credit Type



Hopper Device ID: 1

Number of Coins to Dispense: 1

Credit Type: Cashable

Coin To Dispense: $0.25 USD

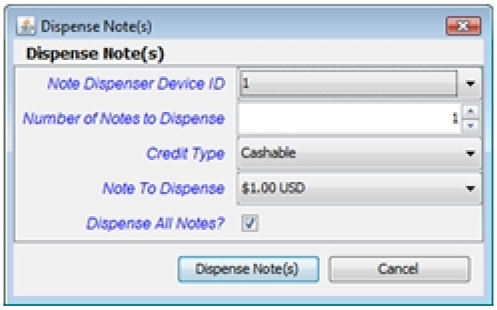
Dispense All? ☑

8. Click **Dispense Coin(s)**.
9. Notice the Credit Meter changes.
   Depending on the credit type you choose to dispense: the **[Cashable]**, **[Promo]** or **[Non-Cash]** field under EGM Status decrements in the amount specified.

Dispense Note(s)

Simulate an EGM dispensing paper currency.

1. Add $10.00 to the Cashable Credit Meter. See Insert Notes.
2. From the Player Verbs tab, click **Dispense Note(s)**.
2. Configure the Dispense Note(s) options as needed.
    - **Note Dispenser Device ID** - Select the identifier of the note dispenser device.
    - **Number of Notes to Dispense** - Type or select the number of notes you want dispensed.
    - **Credit Type** - Select whether the credits dispensed are cashable, non-cashable or promotional.
    - **Note To Dispense** - Select the note denomination to be dispensed.
      Denominations are defined in the G2S Note Acceptor Device section of the `smart-egm.xml` file.
    - **Dispense All Notes?** - If selected, all credits are dispensed, regardless of their classification. This field overrides values set in the the following fields:
        - Number of Notes to Dispense
        - Credit Type



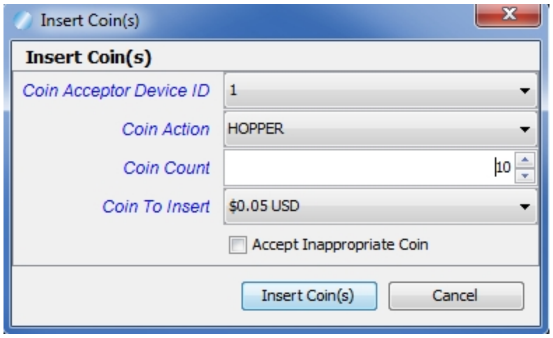
Note Dispenser Device ID: 1
Number of Notes to Dispense: 1
Credit Type: Cashable
Note To Dispense: $1.00 USD
Dispense All Notes? ☑

3. Click **Dispense Note(s)**.
4. Notice the Credit Meter changes. Look under **EGM Status** > credit type field: [Cashable], [Promo] or [Non-Cash] to see the specified amount decrement.

Insert Coin(s)

Simulate inserting coins or tokens into an EGM.

1. Ensure the coin acceptor is enabled.
2. Ensure EGM State is enabled. If not, click **Reset EGM**.
3. From the Player Verbs tab, click **Insert Coin(s)**.



Coin Acceptor Device ID: 1
Coin Action: HOPPER
Coin Count: 10
Coin To Insert: $0.05 USD
□ Accept Inappropriate Coin

4. Configure the Insert Coin(s) options as needed.
   - **Coin Acceptor Device ID**- Select the identifier of the coin acceptor device.
   - **Coin Action** - Select where the coin goes once the coin is inserted: drop or hopper.
     The coin acceptor only accepts denominations that are enabled in the coin acceptor device. If you select **Drop** or **Hopper**, the Credit Meter [Cashable] field increments in the amount specified.
   - **Coin Count** - Type or select the number of coins you want to insert.
   - **Coin To Insert** - Select the coin denomination you want to insert.
     - Coin denominations are defined in the G2S Coin Acceptor Device section of the `smart-egm.xml` file.
     - The coin acceptor only accepts denominations that are enabled in the coin acceptor device.
   - **Accept Inappropriate Coin** - If checked, the RST generates a `G2S_CAE105` (Inappropriate Coin Not Returned) event.

6. Click **Insert Coin(s)**.
7. Look at the Credit Meter, Cashable field (top left of screen) to see the amount incremented.

Insert ID

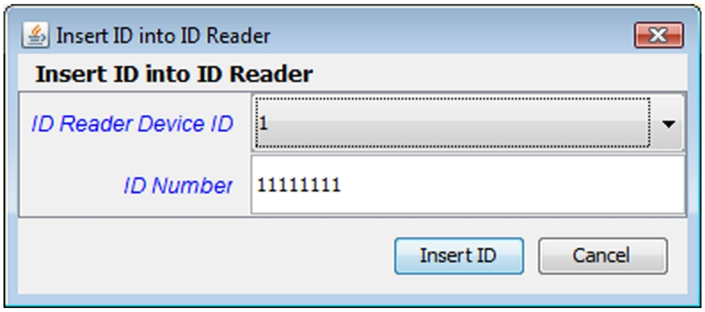
Simulate inserting an employee or a player identifier (such as a card) into an EGM's card reader.

The `id-rst.xml` database is

- Created when you install RST, and automatically provides a randomly selected ID number from the `id-rst.xml` file.
- Used with the SmartEGM user interface only. It is separate from the databases used with Tiger verbs.

Follow these steps to insert a card into the EGM's card reader.

1. From the Player Verbs tab, click **Insert ID.**
2. Configure the *Insert ID into ID Reader* options as needed.
   - **ID Reader Device ID** -Select the device identifier of the ID reader.
   - **ID Number** - Type the player or employee identifier.
     By default, the RGS database is populated with the following IDs:
     - 22222222
     - 12345678
     - 11111111



Insert ID into ID Reader

Insert ID into ID Reader

ID Reader Device ID: 1

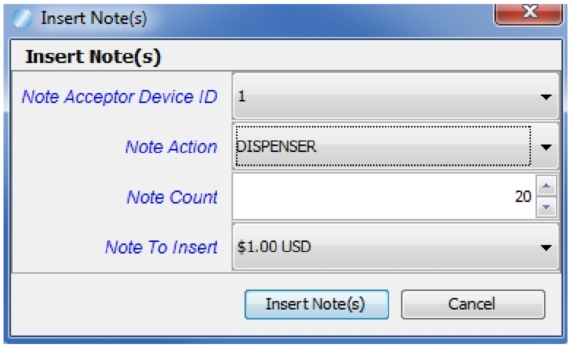
ID Number: 11111111

3. Click **Insert ID**.
4. Look to the Player Display panel (bottom of the screen) and see The Card Reader Bezel field now shows: **Player - [card number]**.

Insert Note(s)

Simulate inserting paper currency into an EGM.

1. From the Player Verbs tab, click **Insert Note(s)**.
2. Configure the Insert Note(s) options as needed.
   - **Note Acceptor Device ID** - Select the identifier of the note acceptor device.
   - **Note Action** - Select where the note goes once inserted. If you select
     - **Drop** or **Dispenser** the **Credit Meter [Cashable]** field increments in the amount specified.
     - **Reject** or **Return**, locks the EGM from accepting notes.
   - **Note Count** - Type or select the number of notes to insert.
   - **Note To Insert** - Select the note denomination. Only active denominations are displayed. Note denominations are defined in the *G2S Note Acceptor Device* section of the `smart-egm.xml` file.



Insert Note(s)

Insert Note(s)

Note Acceptor Device ID: 1

Note Action: DISPENSER

Note Count: 20

Note To Insert: $1.00 USD

3. Click **Insert Note(s)**.
4. You can now simulate: Play game, Perform Note or Coin Drop, Dispense Note(s) and so on.
5. Click the **Transcript** tab to see message flow.

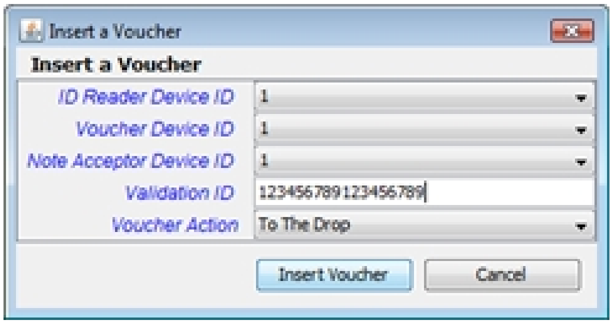
Insert Voucher

Simulate inserting a voucher at an EGM.

1. You must create a voucher before you can insert one.
2. From the Player Verbs tab, click **Insert Voucher**.

   When you insert a voucher, a randomly selected voucher is removed from the voucher database and is used as the default validation ID.
3. Configure the *Insert a Voucher* options as needed.
   - **ID Reader Device ID -** Select the associated device identifier.
   - **Voucher Device ID -** Select the identifier.
   - **Note Acceptor Device ID -** Select the identifier.
   - **Validation ID -** Accept the default validation identifier, or enter an 18-digit number string.
   - **Voucher Action -** Select whether the voucher goes to the drop, is rejected, or returned to the player.

     If you select Reject, the following takes place:
     - The credit meter does not update.
     - `commitVoucher` is sent with transferAmt=0.
     - egmAction=G2S_rejected, egmException=99 (Voucher rejected - reason unknown).
     - The host clears pending redemption so the voucher can be redeemed later.



Insert a Voucher

Insert a Voucher

ID Reader Device ID	1
Voucher Device ID	1
Note Acceptor Device ID	1
Validation ID	123456789123456789
Voucher Action	To The Drop

4. Click **Insert Voucher**.
5. Play a game and/or click the **Transcript** tab to see message flow.

Key Off Handpay

Simulate a key off to enable an EGM after a handpay lockup condition. The actual movement of the credits is initiated by a key switch.

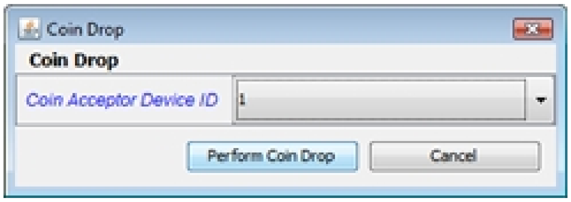
1. From the Player Verbs tab, add cashable amount to Credit Meter. See Insert Note(s).
2. Click **Cash Out to Handpay** to lockup the EGM. See the EGM Status pane:**G2S_egmLocked**.
2. Click **Key Off Handpay** and configure the Key Off options as needed.
   - **Transaction ID** - Select the identifier of the transaction you want to key off.
   - **Key Off Type** - Select how the credits will be paid. to a voucher, the credit meter, a WAT or cash.
   - **Cashable Amount** - Read-only. Amount of cashable funds for the specified transaction ID.
   - **Promo Amount** - Read-only. Amount of promotional funds for the specified transaction ID.
   - **Non Cash Amount** - Read-only. .Amount of non-cashable funds for the specified transaction ID.
3. Click **Key Off**.
4. Look at the **EGM Status** panel.
   - **EGM State** changes to **G2S_enabled**.
   - **Device Class** changes to **G2S_handpay**.
   - **Key Off Timeout**, which displays on the screen when RST locks up, counts down to the key off timeout. You can modify the timeout when you simulate gameplay (Play Simple Game, Play Paytable Game, Play Central Game or Play Progressive Game) by changing the value of the **Key Off Time Out** field value.

     Once the key off timeout reaches zero, the key off handpay lockup is cleared (EGM Status changes back to **G2S_enabled** and the Device Class changes back to **G2S_gameplay**).

Perform Coin Drop

Simulate a coin drop at an EGM.

1. From the Player Verbs tab, click **Perform Coin Drop**.
2. **Coin Acceptor Device ID -** select the identifier of the coin acceptor device.
3. Click **Perform Coin Drop**.
4. Click the **Transcript** tab to see message flow.



Coin Drop

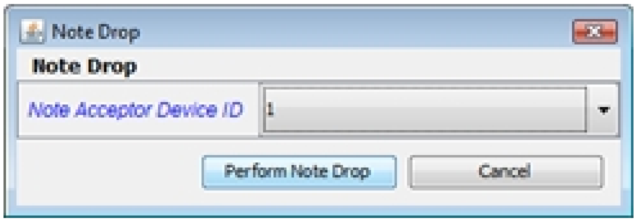
Coin Drop

Coin Acceptor Device ID: 1

Perform Note Drop

Simulate a note drop at an EGM.

1. From the Player Verbs tab, click **Perform Note Drop**.
2. **Note Acceptor Device ID -** select the identifier of the note acceptor device.
3. Click **Perform Note Drop**.
4. Click **Transcript** tab to see the message flow.



Note Drop

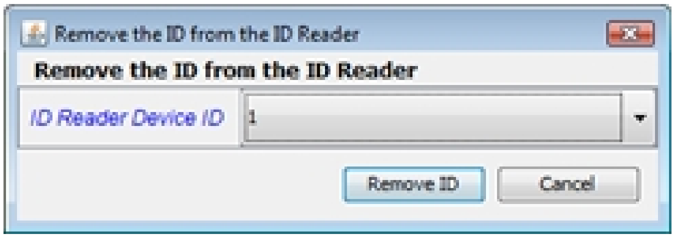
Note Drop

Note Acceptor Device ID: 1

Remove ID

Simulate removing an employee or player identifier (such as a card) from an EGM's card reader.

1. From the Player Verbs tab, click **Remove ID**.
2. **ID Reader Device ID -** select the device identifier of the ID reader from which you want to remove an ID.
3. Click **Remove ID**.
4. Notice the Player Display panel (bottom of the screen) Card Reader Bezel field shows: **Uncarded**.



Remove the ID from the ID Reader

Remove the ID from the ID Reader

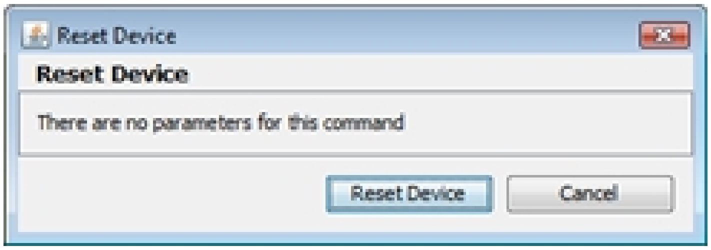
ID Reader Device ID: 1

Reset EGM

Reset the cabinet status of the EGM to the default (enabled). This is not a toggle to reset to previous state, it resets to enabled. Use when you want to "restart" the EGM.

If there is any money on the EGM when it is reset, that money remains on the EGM.

1. From the Player Verbs tab, click **Reset EGM**.
2. Click **Reset Device**.
3. Notice the EGM Status (left panel of interface) the **EGM State**: changes from unlocked to **enabled**.



There are no parameters for this command

About Progressives

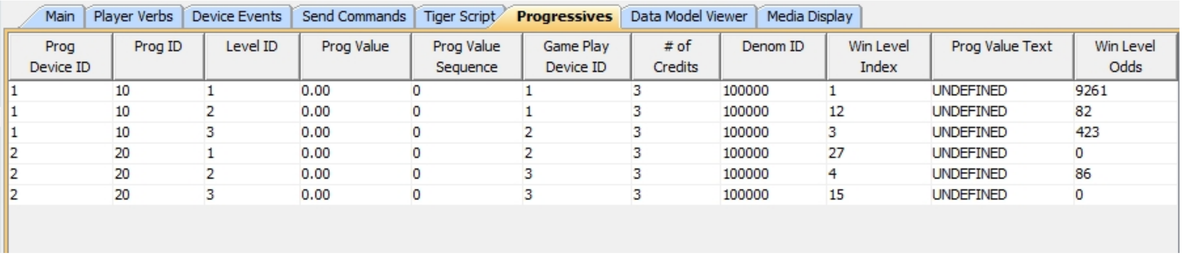
RST provides simulation of traditional progressives—awards that increment based on wagered amounts and on win combinations. Each progressive device:

- Is metered separately.
- Has a unique identifier.
- Is disabled if the EGM loses communication with the controller.
- Has all types of commands including: `deviceID`, `progID` and `levelId`.

There are two Progressive tables:

- Go to **SmartEGM tab > Player Verbs tab > Play Progressive Game**
  Play Game with Progressive Hit - select a specific device ID (EGM), denom value, wager type, and progressive game to play.
  If you see the Play Game button grayed out some of the time, change the wagers to fit one of the progressive levels from the table.

- Go to **SmartEGM tab > Progressives tab**
  Progressive Table - see all progressive games listed in that EGM available for testing. These are defined via the *SmartEGMConfig* file.



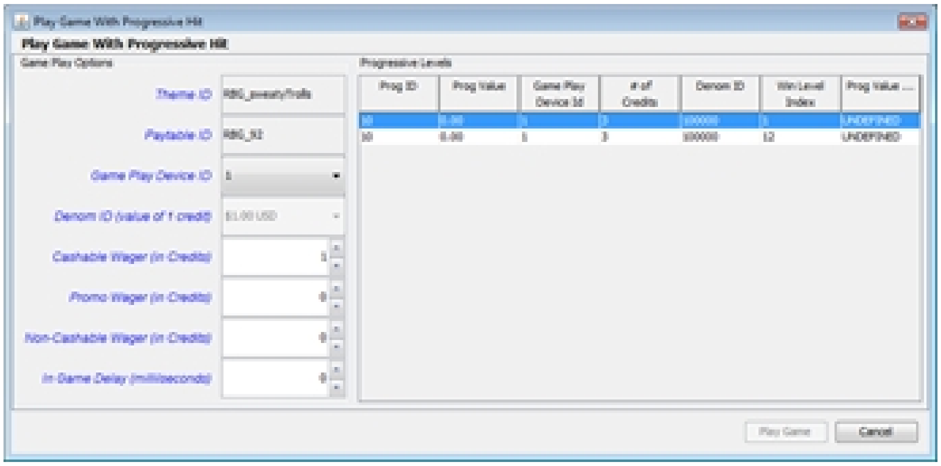
Prog Device ID	Prog ID	Level ID	Prog Value	Prog Value Sequence	Game Play Device ID	# of Credits	Denom ID	Win Level Index	Prog Value Text	Win Level Odds
1	10	1	0.00	0	1	3	100000	1	UNDEFINED	9261
1	10	2	0.00	0	1	3	100000	12	UNDEFINED	82
1	10	3	0.00	0	2	3	100000	3	UNDEFINED	423
2	20	1	0.00	0	2	3	100000	27	UNDEFINED	0
2	20	2	0.00	0	3	3	100000	4	UNDEFINED	86
2	20	3	0.00	0	3	3	100000	15	UNDEFINED	0

You can also request progressive host information through the Send getProgressiveHostInfo option on the Send Commands tab.

Play Progressive Game

Simulate playing a progressive game at an EGM.

1. Add dollars to the **Credit Meter**: Cashable and non-cashable fields.
2. From the **Player Verbs** tab, click **Play Progressive Game**.
   All the following fields are required.
   - **Theme ID** - Read-only field.
     Shows current EGM theme. Dependent on the Game Play Device ID value.
   - **Paytable ID** - Read-only field.
     Shows current EGM paytable identifier. Dependent on the Game Play Device ID value.
   - **Game Play Device ID** - Select the game play device on which you want to play.
   - **Denom ID (value of 1 credit)** - Select the value of a single credit.
   - **Cashable Wager (in Credits)** - Type or select the number of cashable, non-promotional credits to wager.
   - **Non-Cashable Wager (in Credits)** - Type or select the number of non-cashable, non-promotional credits to wager.
   - **Promo Wager (in Credits)** - Type or select the number of promotional credits you want to wager.
   - **In Game Delay (milliseconds)** - Type or select the number of milliseconds to delay when a game completes.



Play Game With Progressive Hit

Play Game With Progressive Hit

Game Play Options

Theme ID: RBG_[illegible]

Paytable ID: RBG_92

Game Play Device ID: 1

Denom ID (value of 1 credit): $1.00 USD

Cashable Wager (in Credits): 1

Promo Wager (in Credits): 0

Non-Cashable Wager (in Credits): 0

In Game Delay (milliseconds): 0

Progressive Levels

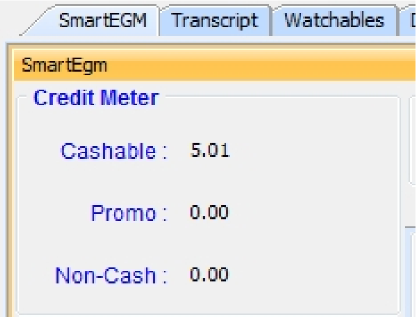
Prog ID	Prog Value	Game Play Device Id	# of Credits	Denom ID	Win Level Index	Prog Value ...
10	0.00	1	3	100000	1	UNDEFINED
10	0.00	1	3	100000	12	UNDEFINED

3. **Progressive Levels** - Click the progressive you want to associate with the win.
4. Click **Play Game**.

About Wagering Account Transfer (WAT)

The WAT sidebar is enabled only if one or more WAT devices are included in the SmartEGM configuration file. See WAT Source Reference for testing code.

When enabled, it shows the movement of cashable, promo, or non-cashable information being transferred either to or from a player's account; which is displayed in the **SmartEgm Status panel > Credit Meter** section.



Cashable : 5.01

Promo : 0.00

Non-Cash : 0.00

The following tables show default settings in the SmartEGM Configuration File, if connected to RGS.

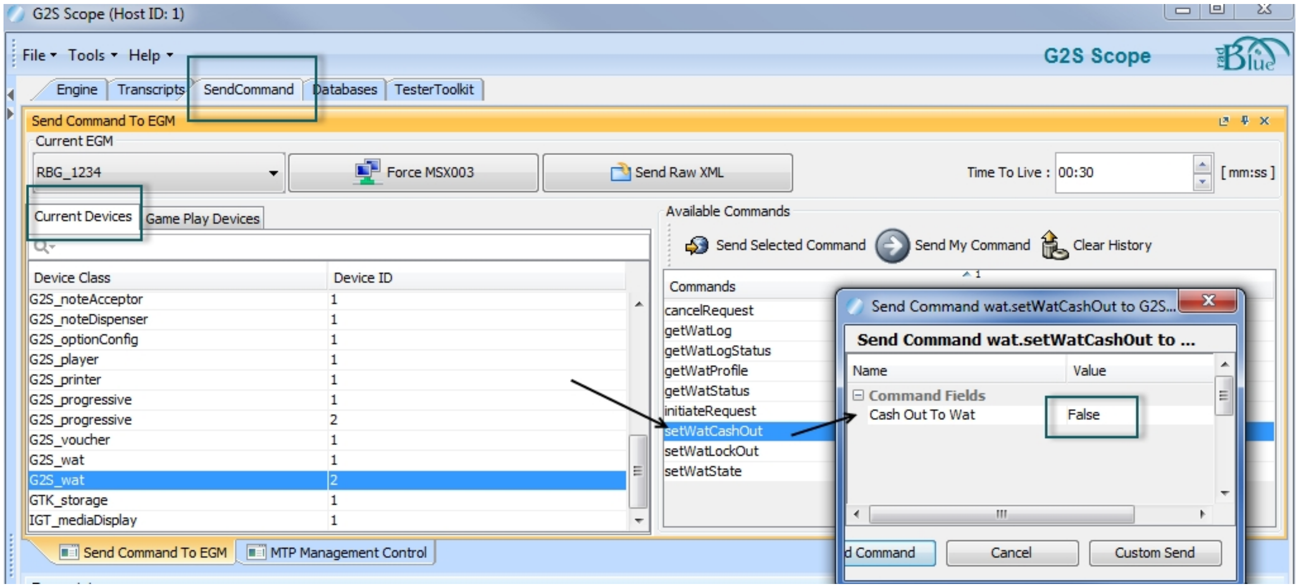
Devices	Description
WAT device 1	represents a host-controlled WAT device (where the host initiates the WAT transfer)
WAT device 2	is an EGM controlled device (where the player initiates the WAT transfer using the EGM’s GUI).
Identifiers	Default Settings for RGS
valid WAT accounts	1111, 2222, 3333, 4444, 5555, 6666, 7777, 8888, 9999, and 12345678
valid player IDs	12345678, 11111111, and 22222222

Configuring WAT

Change WAT parameters on the RST interface.

At the RGS, do the following:

1. Click the **SendCommand** tab.
2. At the **Current Devices** tab, scroll to **G2S_wat**.
3. At the Commands list, select the command to change, in this example **setWatCashOut**.
4. Double-click the **command** to open the pop-up box.
5. Click the **value**, in this case it was True, we changed to **False**.
6. Click **Send Command** to set the value.
7. Go to the **RST > WAT** screen. See the RST Interface.



Current EGM: RBG_1234

Time To Live : 00:30 [mm:ss]

Device Class	Device ID
G2S_noteAcceptor	1
G2S_noteDispenser	1
G2S_optionConfig	1
G2S_player	1
G2S_printer	1
G2S_progressive	1
G2S_progressive	2
G2S_voucher	1
G2S_wat	1
G2S_wat	2
GTK_storage	1
IGT_mediaDisplay	1

Commands
cancelRequest
getWatLog
getWatLogStatus
getWatProfile
getWatStatus
initiateRequest
setWatCashOut
setWatLockOut
setWatState

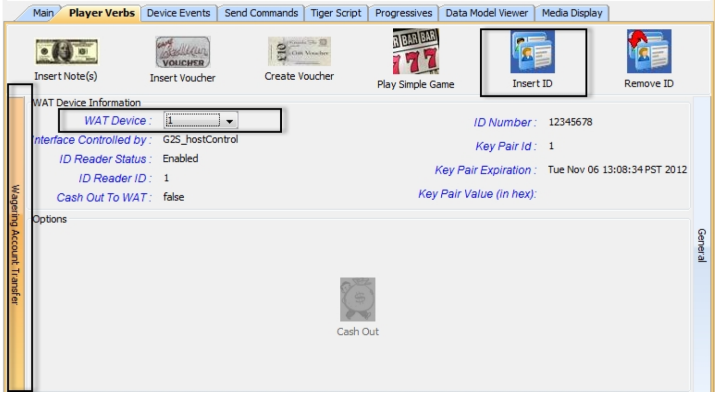
Send Command wat.setWatCashOut to ...

Name	Value
Command Fields	
Cash Out To Wat	False

Adding Promo/Non-Cashable Credits to the Credit Meter Overview

In the top left corner of the RST user-interface you see the Credit Meter section listing: Cashable, Promo and Non-Cash. To add Cashable credits to these meters, use Insert Note(s).
Follow these steps to add Promo and Non-Cash credits.

1. Go to **Player Verbs > WAT** screen.
2. Click **Start SmartEGM** to initiate communication with the host and activate verbs.
3. Click **Insert ID** to insert a player-designated card. For RGS host, use player number 12345678.
4. Click the WAT sidebar and select the **WAT Device**.

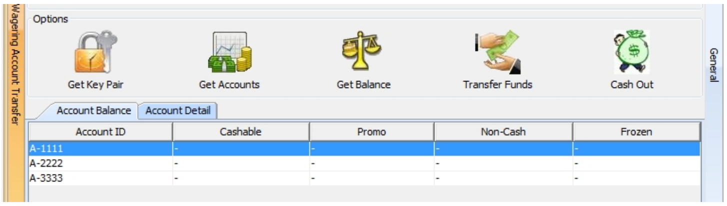


WAT Device Information

WAT Device : 1
Interface Controlled by : G2S_hostControl
ID Reader Status : Enabled
ID Reader ID : 1
Cash Out To WAT : false

ID Number : 12345678
Key Pair Id : 1
Key Pair Expiration : Tue Nov 06 13:08:34 PST 2012
Key Pair Value (in hex):

4. At the WAT screen,
   - Click **Get Key Pair** to initiate WAT messaging. then click **Get Key Pair**.
   - Click **Get Accounts** to open a box, and then click **Get WAT Accounts.**



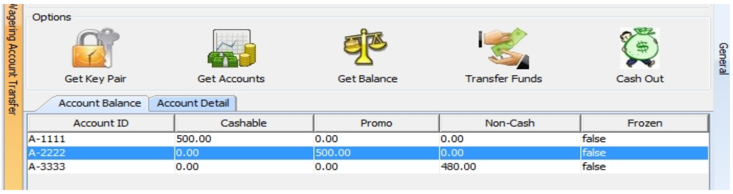
Account ID	Cashable	Promo	Non-Cash	Frozen
A-1111	-	-	-	-
A-2222	-	-	-	-
A-3333	-	-	-	-

5. Highlight the first account. Click **Get Balance**. and click **Get WAT Balance**.
6. Highlight the next account and click **Get Balance** and click**Get WAT Balance**.
7. Highlight athe next account and repeat the steps.

Transfer Funds

Notice the amounts in each column. Follow these steps to add **Promo** funds to the Credit Meter.

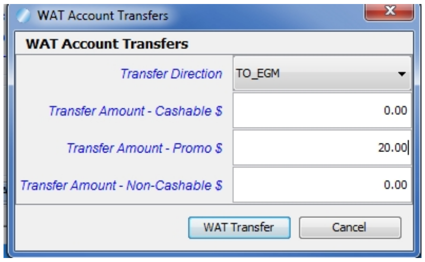
1. Highlight the **Promo** account.



Account ID	Cashable	Promo	Non-Cash	Frozen
A-1111	500.00	0.00	0.00	false
A-2222	0.00	500.00	0.00	false
A-3333	0.00	0.00	480.00	false

2. Click **Transfer Funds**.
   - Transfer Direction, select **to_EGM**
   - Transfer Amount – Promo, type **20.00**.

- Click **WAT Transfer**.



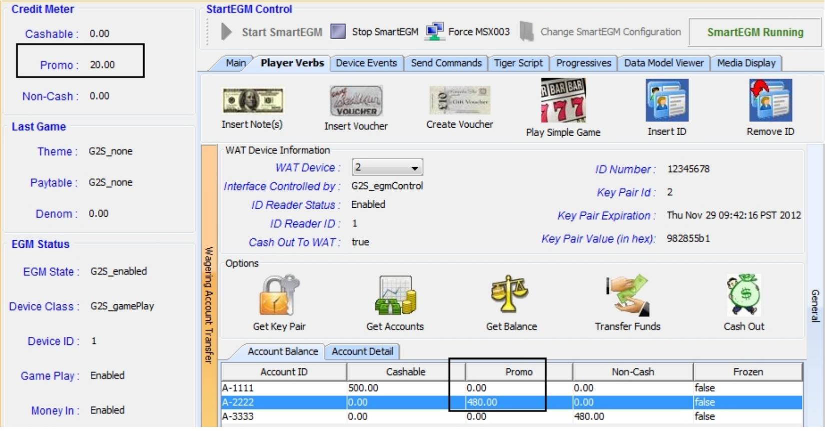
Transfer Direction TO_EGM

Transfer Amount - Cashable $ 0.00

Transfer Amount - Promo $ 20.00

Transfer Amount - Non-Cashable $ 0.00

3. Look at the **Credit Meter** (top left of RST screen). Notice there is now $20.00 listed in the Promo field.
4. Click **Get Balance** to show the new Promo balance for the Promo account. Notice there is now $480.00 in the Promo column.
5. Repeat steps to add **Non-Cashable** to Credit Meter.



Credit Meter

Cashable : 0.00

Promo : 20.00

Non-Cash : 0.00

Last Game

Theme : G2S_none

Paytable : G2S_none

Denom : 0.00

EGM Status

EGM State : G2S_enabled

Device Class : G2S_gamePlay

Device ID : 1

Game Play : Enabled

Money In : Enabled

WAT Device Information

WAT Device : 2

Interface Controlled by : G2S_egmControl

ID Reader Status : Enabled

ID Reader ID : 1

Cash Out To WAT : true

ID Number : 12345678

Key Pair Id : 2

Key Pair Expiration : Thu Nov 29 09:42:16 PST 2012

Key Pair Value (in hex): 982855b1

Account ID	Cashable	Promo	Non-Cash	Frozen
A-1111	500.00	0.00	0.00	false
A-2222	0.00	480.00	0.00	false
A-3333	0.00	0.00	480.00	false

About Source Reference

The RST provides Source Reference options to test your audit/reconciling transactions for EGM events.

These options show you how to generate tests to use on the RST— to ensure that code is working as designed. The end result is to validate that one log points to the appropriate log entry.

The following examples show you how to generate transactions to test source-reference code for reconciling transactions on an EGM:

- Generating a handpay
- Generating a voucher
- Generating a WAT

About Source Reference

The RST provides Source Reference options to test your audit/reconciling transactions for EGM events.

These options show you how to generate tests to use on the RST— to ensure that code is working as designed. The end result is to validate that one log points to the appropriate log entry.

The following examples show you how to generate transactions to test source-reference code for reconciling transactions on an EGM:

- Generating a handpay
- Generating a voucher
- Generating a WAT

Generating a Handpay Source Reference

Follow these steps to see the added handpay source reference in RST:

1. From the Player Verbs tab on the SmartEGM layout, click **Insert Note** and insert a 1-dollar note into EGM.
2. Click **Play Simple Game**.
3. Type **1** in the **Primary Win (in Credits)** field.
4. Select **Win to handpay**.
5. Click **How to pay handpay** and select **Pay to Handpay**.
6. Click **Play Game**.
7. Select the Transcript tab, and review the `handpay.handpayRequest` command to see the **handpaySourceRef**. For example:

```
<g2s:handpaySourceRef g2s:cashableAmt="1000" g2s:deviceClass="G2S_
gamePlay" g2s:deviceId="1"

g2s:logSequence="8"
 g2s:nonCashAmt="1000"
 g2s:promoAmt="1000"
         g2s:transactionId="27"/>
```

Generating a Voucher Source Reference

Follow these step to see the added voucher source reference in RST:

1. From the Player Verbs tab on the SmartEGM layout, click **Insert Note** and insert a 1-dollar note into EGM.
2. Click **Cash Out to Handpay**.
3. Click **Key Off Handpay**.
4. In the **Key Off Type** field, select **Pay to Voucher**.
5. Click **Key Off**.
6. Select the Transcript tab, and review the `voucher.issueVoucher` and `voucher.voucherLogList` commands to see the **voucherSourceRef**. For example:

```
<g2s:voucherSourceRef g2s:cashableAmt="100000"
g2s:deviceClass="G2S_handpay" g2s:deviceId="1"
```

```
g2s:logSequence="2"
g2s:nonCashAmt="0"
g2s:promoAmt="0"
g2s:transactionId="3"/>
```

Generating a WAT Source Reference

Follow these steps to see the added WAT source reference in RST:

1. From the SmartEGM layout, select the **WAT Transfers** tab.
2. Click the **WAT Device** and select an EGM-controlled WAT device. If the selected WAT device is EGM-controlled, additional options (such as Get Key Pair) display in the Options sections.
3. Click **Get Key Pair**, and then click **Get Key Pair**.
4. Click **Insert ID**, enter a valid player ID (**12345678**), and **click Insert ID**.
5. From the RGS (or your host system), send the `WAT.setWatCashOut` command with a *cashOutToWat* value of true. (In RGS, go to: **Send Command** > **G2S_wat** > **WAT - Set WAT Cash Out**)
6. From the RST SmartEGM Player Verbs tab, click Insert Note and insert a 1-dollar note into EGM.
7. Click **Play Simple Game**.
8. Type **1** in the **Primary Win (in Credits)** field.
9. Select **Win to handpay**.
10. Click the **How to pay handpay** and select **Pay to WAT**.
11. Click **Play Game**.

12. Select the Transcript tab, and review the `wat.commitTransfer` command to see the **watSourceRef**. For example:

    ```
    <g2s:watSourceRef g2s:cashableAmt="1000" g2s:deviceClass="G2S_
    handpay" g2s:deviceId="1"

    g2s:logSequence="5"
    g2s:nonCashAmt="0"
    g2s:promoAmt="0"
    g2s:transactionId="21"/>
    ```

    Note: If RST receives a `wat.error` in response to an `initiateTransfer` request, it commits the transfer.

Using Device Events Overview

Follow these steps to simulate device events.

1. Click **Start SmartEGM** on the SmartEGM layout to initiate communication with the host and enable Device Events options.
2. Click **Device Events**.
3. Select the device class for the events you want to send from the Classes list on the left side of the screen. The options for the selected class open in the right-hand panel.
4. Select the device. If the device list is disabled, there is only one device for the selected class.
5. Select **command** or **event** to send to host. Options vary depending on the class:
   - cabinet
   - coin acceptor
   - hardware
   - note acceptor
   - printer
6. Select the **Transcript** tab to see the message flow.

   When you select a request message, the corresponding response is highlighted. Double-click the message to view its details.

Cabinet Events

You can simulate Cabinet events using the following two menus:

- **Standard Menu** - Tests the cabinet. Simulate errors or malfunctions associated with the following physical devices: a cabinet, a coin acceptor, a note acceptor, or a printer.
  Events (such as turning on/off a service button, lamp, and video display) to the physical housing or check security for a specified cabinet device.
- **Operator Menu** - to simulate an employee at the EGM. Send commands (enable, disable [due to tilt], and lock [the EGM from play]) to the host as soon as you select the associated option.

The host must subscribe to the events so the messages are received.

Cabinet Events Using the Standard Menu

Standard Menu

Use the standard menu to simulate cabinet events. The following are grouped by cabinet event similarities:

- **Toggle Events** - Events that change from one condition to another.
- **Mechanical Errors** - Events that lock-up a game.
- **Continued Events** - Events that do not lock-up the game.

Toggle Events

Toggle events change from one condition to another. The steps for these three events are similar. See **Test a Service Lamp** for the steps.

- Service Lamp Off to On
- Backup Battery Low to Backup Battery
- EGM Power Up to Power Lost

Test a Service Lamp

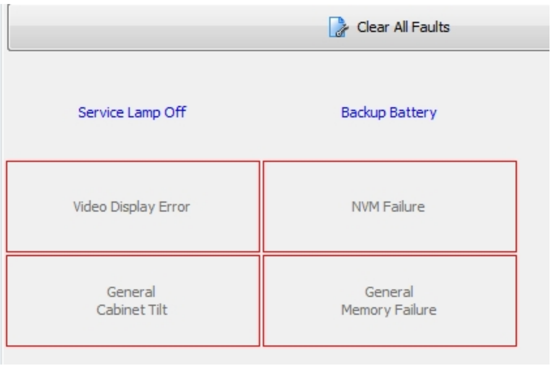
Follow these steps to simulate testing a service lamp.

1. Select the **Cabinet Device**, if this field is activated.
2. Click **Service Lamp Off**. Once clicked it toggles to Service Lamp On.
3. Click the RST **Transcript** tab to see the commands sent.
   - eventReport: G2S_CBE301 - Service Lamp On
   - eventReport: G2S_CBE302 - Service Lamp Off
4. Use the (Tilt Events) Cabinet Door to clear events by clicking **opening** and then **closing**:
   - eventReport: G2S_CBE307 - Cabinet Door Open
   - eventReport: G2S_CBE308 - Cabinet Door Closed
   - eventReport: G2S_CBE002 - EGM Enabled Cabinet
   - eventReport: G2S_CBE205 - EGM Enabled and PlayableEGM Enabled and Playable

Mechanical Errors Events

At **Cabinet Events** > **Standard** menu, there are four events that simulate mechanical errors that lock-up a game, causing the SmartEGM file to automatically disable game-play.

- Video Display Error
- NVM Failure (non-volatile memory)
- General Cabinet Tilt
- General Memory Failure



1. Click **Video Display Error**. Selected events are framed in red.
2. Look at the **EGM Status** (left panel of the screen) and notice the EGM State changes from `G2S_enabled` to `G2S_egmDisabled`.
3. Click the **Transcript** tab to see the messages.
4. Click **Clear All Faults** before you can use them again.
   For Door events, click **Closed** to clear the faults.

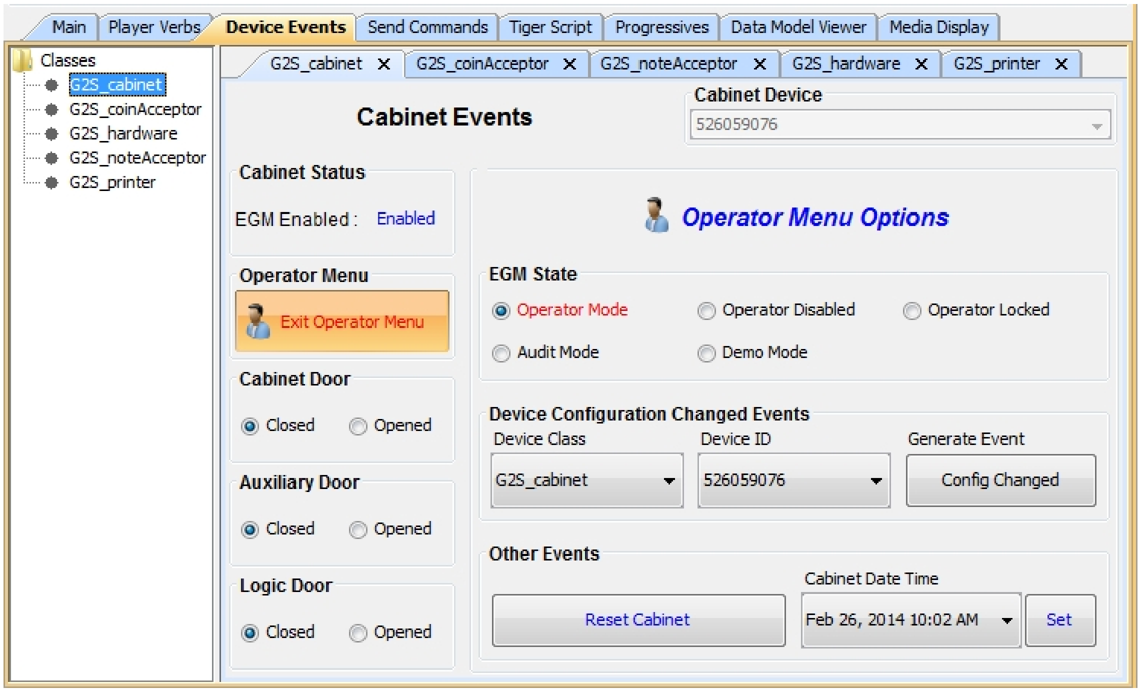
Continued Events

Click any of the remaining events, see the Transcript, and continue to use them without clearing the fault. These events do not disable the EGM from game play.

Cabinet Events Using the Operator Menu

The Operator Menu lets you simulate an employee at the EGM. In this mode you can:

- Enable an EGM and get it ready for play.
- Clear an operator locked status.
- Initiate demo mode.
- Change the device configuration.
- Reset the cabinet date and time.



Cabinet Device: 526059076

Cabinet Status
EGM Enabled : Enabled

Operator Menu

Cabinet Door: ● Closed ○ Opened

Auxiliary Door: ● Closed ○ Opened

Logic Door: ● Closed ○ Opened

EGM State
● Operator Mode ○ Operator Disabled ○ Operator Locked
○ Audit Mode ○ Demo Mode

Device Configuration Changed Events
Device Class: G2S_cabinet
Device ID: 526059076
Generate Event

Other Events
Cabinet Date Time: Feb 26, 2014 10:02 AM

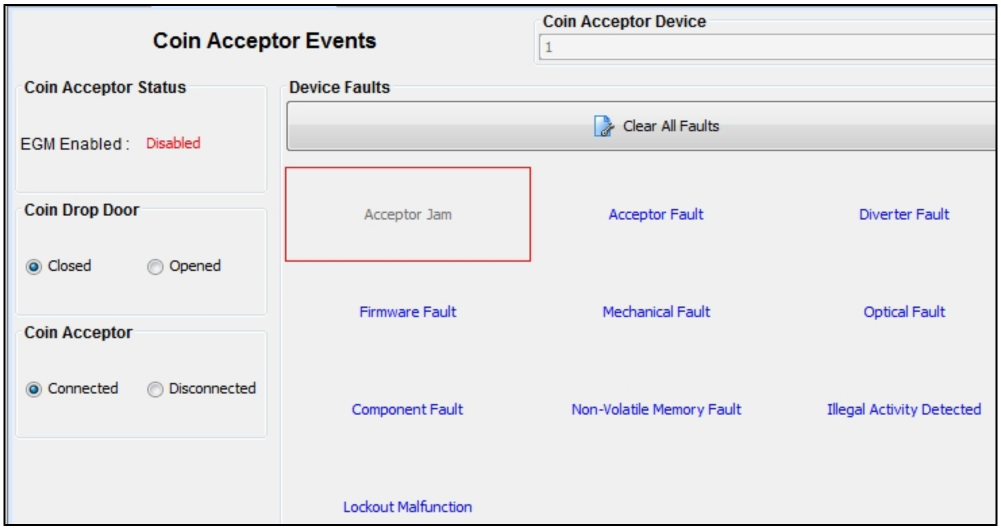
Operator Menu Descriptions

EGM State	Description	Sends the G2S_[message] to the host
Enter Operator Menu	To enter Operator Menu Changes EGM Status at the host from `G2S_operatorDisabled` to `G2S_operatorMode`	`G2S_CBE206` (Operator Menu Activated)
Exit Operator Menu	To exit Operator Menu. Changes EGM status at the host to `G2S_operatorDisabled` and returns to the Standard menu.	`G2S_CBE205` (EGM Enabled and Playable)
Operator Disable	Changes EGM status at the host to `G2S_operatorDisabled` and returns to the Standard menu.	`G2S_CBE202` (EGM Enabled-Operator Menu)
Operator Locked	Changes EGM status at the host to `G2S_operatorLocked` and returns to the Standard menu. Click **Enter Operator Menu** to clear the `G2S_operatorLocked` status.	`G2S_CBE209` (EGM Locked - Operator Menu)
Audit Mode	Change the EGM status at the host to `G2S_auditMode`	`G2S_CBE208` (Meter/Audit Mode Initiated)
Demo Mode	Changes EGM to demonstration mode by sending `G2S_egmDisabled` and returns to the Standard menu. Click **Enter Operator Menu** clears the `G2S_operatorLocked` status.	`G2S_CBE207` (Demo Mode Activated)
Device Configuration Changed Event	Notifies the host of a change at the EGM after you select a device ID or click **Config Changed** button.	`G2S_CBE006` (Operator Changed Cabinet Config)

Coin Acceptor Events

This option sends one or more coin acceptor events for a specified coin acceptor device. All coin acceptor events cause mechanical errors that disable the EGM, see EGM State (EGM).

1. From the SmartEGM layout, click the **Device Events** tab.
2. From the Tree, click **G2S_coinAcceptor**.



Coin Acceptor Events

Coin Acceptor Device: 1

Coin Acceptor Status

EGM Enabled : Disabled

Coin Drop Door: ● Closed ○ Opened

Coin Acceptor: ● Connected ○ Disconnected

Device Faults

3. Select the **Coin Acceptor Device**. If the Coin Acceptor Device field is disabled, there is only one defined device for this class.
4. Select the event you want to simulate.

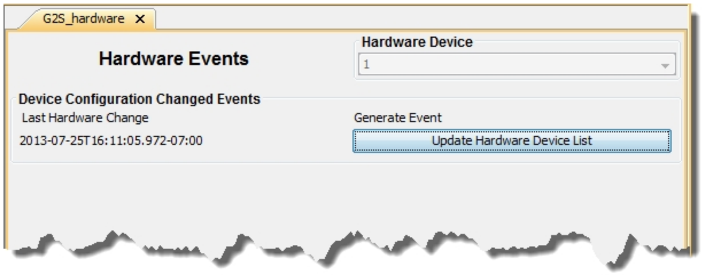
   Selected events are framed in red. You can also toggle Coin Drop Door Closed/Open and Coin Acceptor Connected/Disconnected.

   Note that events are sent to the host as soon as you select the associated option.
5. Click **Transcript** tab to see the message flow.
6. Click **Clear All Faults** to send a **G2S_PTE000 - All Device Faults Cleared** event. An event is then sent for each device that had a fault that indicates the device has been enabled.

Hardware Device Events

This option sends the G2S_HWE100 - Hardware Device List Updated event to the host.

1. From the SmartEGM layout, click the **Device Events** tab.
2. From the Tree, click **G2S_hardware**.



Hardware Events

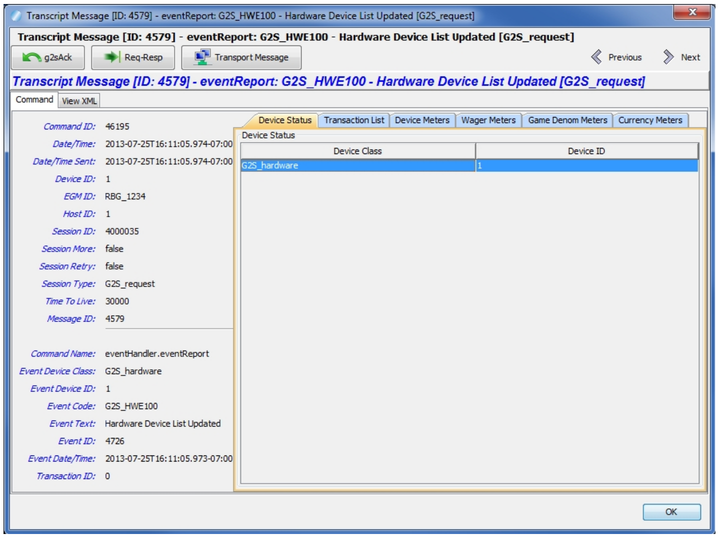
Hardware Device
1

Device Configuration Changed Events

Last Hardware Change
2013-07-25T16:11:05.972-07:00

Note: The hardware class is a single device class, so the Hardware Device field is disabled.

3. Click **Update Hardware Device List**.
4. Click the **Transcript** tab to see the `G2S_HWE100` event and the host system response.



Transcript Message [ID: 4579] - eventReport: G2S_HWE100 - Hardware Device List Updated [G2S_request]

Transcript Message [ID: 4579] - eventReport: G2S_HWE100 - Hardware Device List Updated [G2S_request]

Transcript Message [ID: 4579] - eventReport: G2S_HWE100 - Hardware Device List Updated [G2S_request]

Command ID:	46195
Date/Time:	2013-07-25T16:11:05.974-07:00
Date/Time Sent:	2013-07-25T16:11:05.974-07:00
Device ID:	1
EGM ID:	RBG_1234
Host ID:	1
Session ID:	4000035
Session More:	false
Session Retry:	false
Session Type:	G2S_request
Time To Live:	30000
Message ID:	4579
Command Name:	eventHandler.eventReport
Event Device Class:	G2S_hardware
Event Device ID:	1
Event Code:	G2S_HWE100
Event Text:	Hardware Device List Updated
Event ID:	4726
Event Date/Time:	2013-07-25T16:11:05.973-07:00
Transaction ID:	0

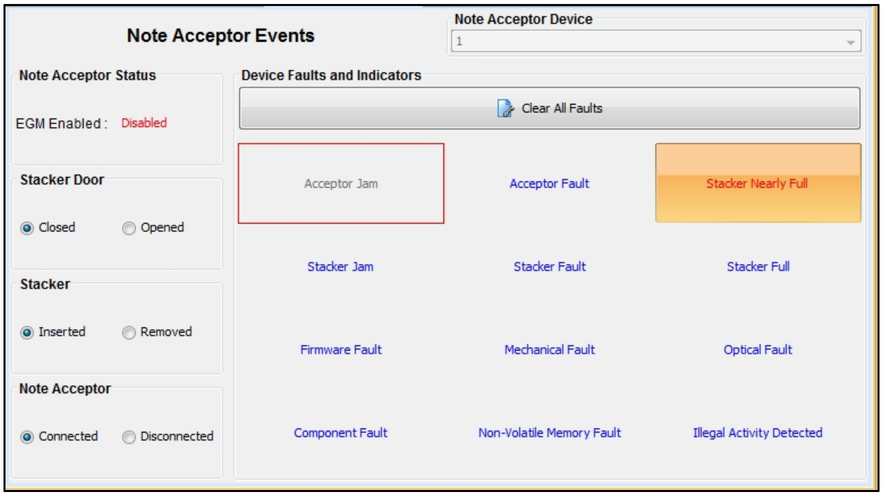
Device Status

Device Class	Device ID
G2S_hardware	1

Note Acceptor Events

This option sends one or more note acceptor events for a specified note acceptor device. All note acceptor events, except Stacker Nearly Full, are errors that disable the EGM.

1. From the SmartEGM layout, click the **Device Events** tab.
2. From the Classes tree, click **G2S_noteAcceptor.**



Note Acceptor Events

Note Acceptor Device: 1

Note Acceptor Status

EGM Enabled : Disabled

Stacker Door

● Closed ○ Opened

Stacker

● Inserted ○ Removed

Note Acceptor

● Connected ○ Disconnected

3. Select a **Note Acceptor Device,** if applicable.
4. Select the **events**. Selected events are framed in red. Events are sent to the host as soon as you select the associated option.
5. Click **Transcript** tab to see message flow.
6. Click **Clear All Faults**.

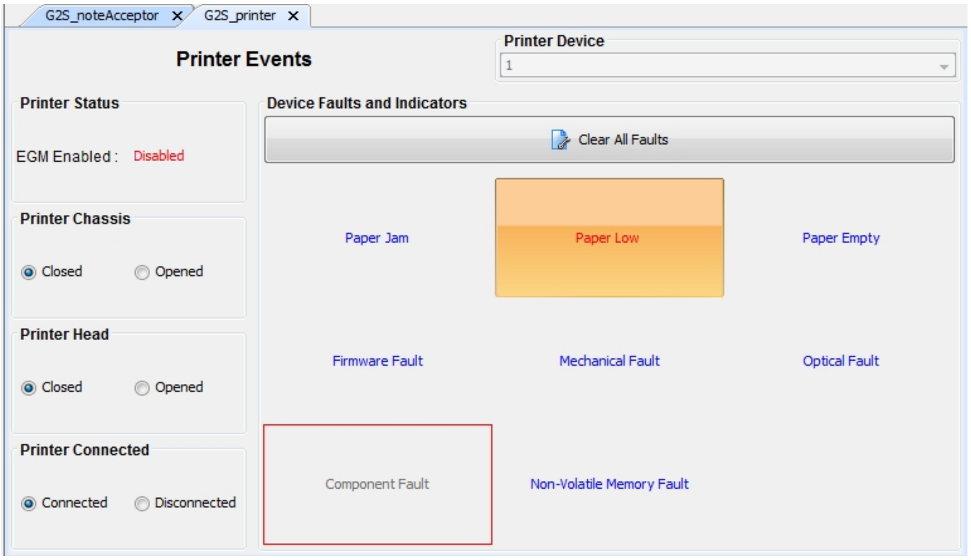
   Note: To clear a **Stacker Nearly Full** event, go to *Stacker*, click **Removed** and then click **Inserted**.

Printer Events

This option sends one or more printer events for a specified printer device. All events except Paper Low are mechanical errors that disable the EGM, see EGM State (EGM).

1. From the SmartEGM layout, click the **Device Events** tab.
2. Click **Printer Events**.
3. Select a **Printer Device** , if activated.
4. Select the **event**.
   Selected events are framed in red.
   Events are sent to the host as soon as you select the associated option.
5. Click **Transcript** tab to see message flow.
6. Click **Clear All Faults**.

Note: **Paper Low** is an indicator and toggles by changing the box colors from yellow to orange.



Printer Events

Printer Device: 1

Printer Status

EGM Enabled : Disabled

Printer Chassis

● Closed ○ Opened

Printer Head

● Closed ○ Opened

Printer Connected

● Connected ○ Disconnected

Device Faults and Indicators

About Send Commands

From the Send Commands tab, you can send EGM-initiated commands to:

- `optionList`
- `commsHostList`
- `getMcastKeyUpdate`
- `getCountdownOverride`

Except `getMcastKeyUpdate`, the steps are similar for the rest of the commands. They follow Using Send Command Overview.

Using Send Commands Overview

Follow these steps to send the `optionList`, `commsHostList`, or `getCountdownOverride` commands:

1. Click **Start SmartEGM** to initiate communication with the host and enable Send Command options.
2. Go to **SmartEGM layout** tab **> Send Commands** tab**.**
3. Select the **command** to send to the host.
4. Select the **device**.
5. Click **Send**.
6. At the confirmation box, click **OK**.
7. Look at the host **Transcript** tab for message flow.

   If the RST is connected to an RGS, click the **Transcripts** tab; if connected to your network, check host transcripts.

Send optionList

Send this command from an EGM to the host for a list of EGM changes to its current configuration.

Follow Using Send Commands Overview to use this option.

Send commsHostList

Send this command from an EGM to the host for the current list of descriptors.

Follow Using Send Commands Overview to use this option.

Send getCountdownOverride

Send this command from an EGM to the host for the current active countdown override from a specified player device.

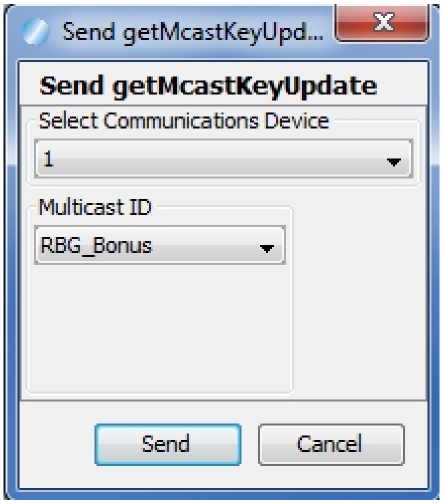
Follow Using Send Commands Overview to use this option.

Send getMcastKeyUpdate

Send this command from an EGM to the host so the SmartEGM can update its security parameters.

Follow these steps to use this option.

1. Click **Start SmartEGM.** to initiate communication with the host and enable Send Command options.
2. Go to **SmartEGM layout** tab **> Send Commands** tab.
3. Select the **communications device**.
4. **Multicast ID** - select a target multicast group.
5. Click **Send** to send the `communications.getMcastKeyUpdate` command.
6. At the confirmation box, click **OK**.
7. Look at the host **Transcript** tab for message flow.



Send getMcastKeyUpd...

Send getMcastKeyUpdate

Select Communications Device: 1

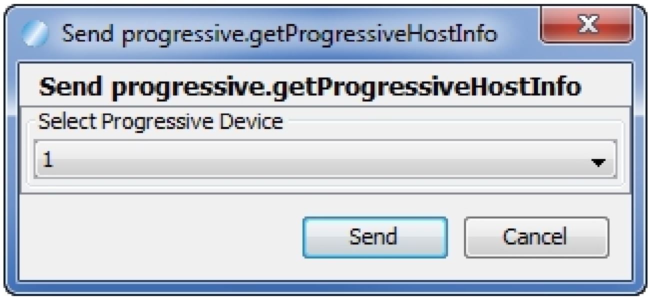
Multicast ID: RBG_Bonus

Send Get Progressive Host Info

Simulate sending an EGM command to a progressive host querying progressive identifiers and levels of support.

The EGM sends a `progressive.getProgressiveHostInfo` command to the host, requesting information on the specified host. The host returns with `progressive.progressiveHostInfo` command.

1. From the Send Commands tab, click **Send getProgressiveHostInfo**.

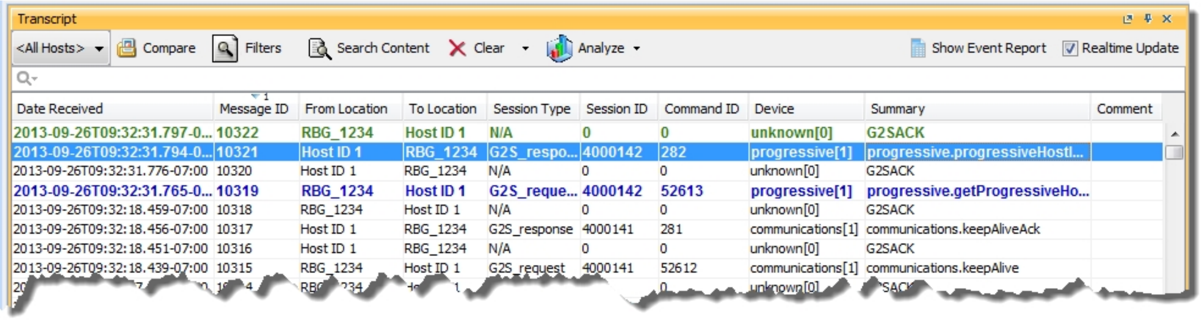


Send progressive.getProgressiveHostInfo

Send progressive.getProgressiveHostInfo

Select Progressive Device: 1

2. Select the identifier of the progressive host you want to query.
3. Click **Send**.
4. Go to the **Transcript** layout, and select the **Transcript** tab to view the Message Transcript.
5. Verify that the `progressive.getProgressiveHostInfo` command was sent and a `progressive.progressiveHostInfo` response command was received.



Date Received	Message ID	From Location	To Location	Session Type	Session ID	Command ID	Device	Summary	Comment
2013-09-26T09:32:31.797-0...	10322	RBG_1234	Host ID 1	N/A	0	0	unknown[0]	G2SACK	
2013-09-26T09:32:31.794-0...	10321	Host ID 1	RBG_1234	G2S_respo...	4000142	282	progressive[1]	progressive.progressiveHostI...	
2013-09-26T09:32:31.776-07:00	10320	Host ID 1	RBG_1234	N/A	0	0	unknown[0]	G2SACK	
2013-09-26T09:32:31.765-0...	10319	RBG_1234	Host ID 1	G2S_reque...	4000142	52613	progressive[1]	progressive.getProgressiveHo...	
2013-09-26T09:32:18.459-07:00	10318	RBG_1234	Host ID 1	N/A	0	0	unknown[0]	G2SACK	
2013-09-26T09:32:18.456-07:00	10317	Host ID 1	RBG_1234	G2S_response	4000141	281	communications[1]	communications.keepAliveAck	
2013-09-26T09:32:18.451-07:00	10316	Host ID 1	RBG_1234	N/A	0	0	unknown[0]	G2SACK	
2013-09-26T09:32:18.439-07:00	10315	RBG_1234	Host ID 1	G2S_request	4000141	52612	communications[1]	communications.keepAlive	

To view the details of a message, double-click the row.

About Tiger Scripting

Tiger is a scripting language developed specifically for use with RadBlue tools. This feature is optional and requires a license to use it. Tiger is intended for non-programmers to create robust test scenarios. Tiger scripts:

- Automate testing on your G2S-driven applications.
- Extend sequences of G2S host-initiated commands.
- Run for extended time periods, sending command after command to the EGM.This type of extensive testing is useful for detecting memory leaks and performing regression testing.

With Tiger, human actions (causing a series of actions in our tools) and GSA verbs can be easily expressed. Verbs denote high-level actions, such as inserting notes into an EGM, expressed in the verb `Human.insertNote`. Tiger scripting allows you to:

- Specify the quantity, denomination, currency, and where the notes go (to the drop or the dispenser).
- Use these verbs to send appropriate events that update meters and logs.

You can use one of the sample Tiger scripts available with RST or create your own Tiger scripts. The *Tiger Scripting Reference* shows you how to create a Tiger script, including definitions of all available Tiger verbs and their attributes. It is available through RST Help or on the RadBlue Document Library.

See About the Example Tiger Scripts for more information on the provided Tiger scripts.

About the Example Tiger Scripts

Example Tiger scripts containing the new Tiger verbs are located in the following directory: [**productDirectory**] > **radblue** > **gsa** > **scripts** > **smart-egm**

Example Tiger scripts can be used as templates for creating custom Tiger scripts.

File Name: smartegm-example-databases-001.xml

This Tiger script:

- Verifies that the ID reader, voucher, note acceptor, and communications devices exist and are enabled (waiting up to five minutes for each to be enabled).
- Inserts a player ID from the id-example.xml database.
- Inserts five, one-dollar notes.

- Cashes out the credit meter to a voucher, which is inserted into the voucher-example.xml database.
- Inserts voucher from the voucher database using the default voucher and note acceptor devices.
- Removes ID to the ID database.
- Repeats ten times.

File Name: smartegm-example-databases-002.xml

This Tiger script:

- Verifies that the ID reader, voucher, note acceptor, and communications devices exist and are enabled (again waiting up to five minutes for each).
- Inserts a player ID from the id-example.xml database.
- Inserts 10 one-dollar notes.
- Cashes out the credit meter to a voucher, which is inserted into the voucher-example.xml database. Rrepeats 10 times to create 10 vouchers in the database.
- Inserts 10 vouchers from the voucher database, using the default note acceptor and voucher devices.
- Removes the player ID from the ID reader, clearing the flag in the database.

File Name: smartegm-example-databases-003.xml

This Tiger script:

- Inserts a player ID from the ID database, but does not lock the ID.
- Removes the player ID from the ID reader.
- Repeats script ten times.

Using Tiger Scripts with an ID Database

To use ID database Tiger verbs - `Human.insertIDFromDatabase` and `Human.insertIDToDatabase` - you must create a file of player IDs. These IDs should correspond to the player IDs in your host system.

1. Open an XML file (using either an XML editor or by saving a text file with an .xml extension).
2. Save the file.

   If you are using a RadBlue smartegm-example-database-00X.xml Tiger script, type **id-example.xml** for the file name. Otherwise, the file name should reflect the value of the database-name attribute in the ID database Tiger verbs, in the format:

   id-[database-name].xml

3. Save the file to the following location:
   [**productDirectory**] > **radblue** > **gsa** > **script** > **conf**
4. Enter the following text, substituting the ID numbers for your host's player IDs:

   ```
   <id-database>

       <id number="22222222" lock="false"/>
       <id number="88888888" lock="false"/>
       <id number="12345678" lock="false"/>
       <id number="11111111" lock="false"/>
       <id number="99999999" lock="false"/>

   </id-database>
   ```

   The player ID numbers used in this example are valid if you are using the G2S Scope (RGS).

5. For each ID number, enter “false” for the lock attribute. The lock attribute allows you to indicate that a player ID is in use (inserted into an EGM). The lock attribute is automatically set to “true” when an ID is in use.
6. Save and close the file.

Using Tiger Scripts with a Voucher Database

To use voucher database Tiger verbs - `Human.insertVoucherFromDatabase` and `Human.createVoucherToDatabase` - you must create a file of voucher numbers. These 18-digit numbers should correspond to the voucher numbers in your host system.

The tiger verb `Human.createVoucherToDatabase` is confined to 500 records so it does not interfere with normal disk space, memory, or CPU functions.

If you are using a RadBlue **smartegm-example-database-00X.xml** Tiger script, a voucher-example.xml voucher database file is automatically created and populated with voucher numbers.

If you want to use voucher numbers from your host system, use the following procedure:

1. Open an XML file (using either an XML editor or by saving a text file with an .xml extension).
2. Name the file.

   The file name should reflect the value of the database-name attribute in the voucher database Tiger verbs, in the format:

   id-[database-name].xml

   When you run your custom Tiger script, the voucher numbers from the specified database will be used.

3. Save the file to the following location:
   [**productDirectory**] > **radblue** > **gsa** > **script** > **conf**

4. Enter the following text, substituting the voucher numbers for voucher numbers in your host system's voucher database:

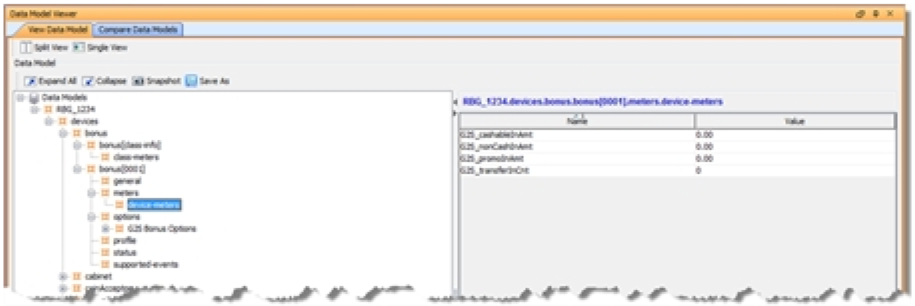
```
<voucher-database>

<id number= "300011224778107571"/>
<id number= "300021224778107572"/>
<id number= "300031224778107573"/>
<id number= "300041224778107574"/>
<id number= "300051224778107575"/>

</voucher-database>
```

5. Save and close the file.

About the Data Model Viewer

As commands are sent from an EGM, the RGS data model changes. For example, if an EGM's meters are incremented, the corresponding meters are updated in the RGS data model when RGS receives the appropriate event message. The data model not updating as expected could indicate an issue with G2S protocol compliance or a general implementation issue.

The RGS Data Model Viewer lets you easily see changes to the RGS data model as they occur. Using a tree-structure control, you can quickly drill-down to the attribute(s) or information you want to view.



RBG_1234.devices.bonus.bonus[0001].meters.device-meters

Name	Value
G2S_cashableInAmt	0.00
G2S_nonCashInAmt	0.00
G2S_promoInAmt	0.00
G2S_transferInCnt	0

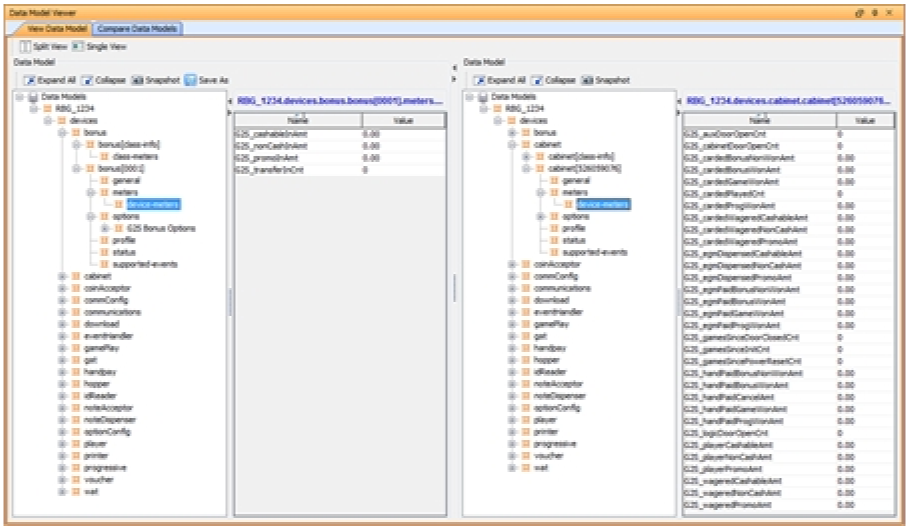
RGS supports up to five EGMs and you can select the data model you want to view. A Split View option lets you view two different areas of a data model at once or view information for two different EGMs at once.

All information displayed in the Data Model Viewer is *read-only* and cannot be modified. However, you can export the selected data model for use with the RadBlue System Tester (RST). Attributes that are deprecated in G2S 2.1 display in a strike-through font (for example, ~~Enable Money Out~~).

Split the Data Model View

Use the Split View option to view two different areas of a data model at once or view information for two different EGMs at once.

1. Go to: **Databases** > **Data Model Viewer**



RBG_1234.devices.bonus.bonus[0001].meters...

Name	Value
G2S_cashableInAmt	0.00
G2S_nonCashInAmt	0.00
G2S_promoInAmt	0.00
G2S_transferInCnt	0

RBG_1234.devices.cabinet.cabinet[526059076...

Name	Value
G2S_auxDoorOpenCnt	0
G2S_cabinetDoorOpenCnt	0
G2S_cardedBonusNonWonAmt	0.00
G2S_cardedBonusWonAmt	0.00
G2S_cardedGameWonAmt	0.00
G2S_cardedPlayedCnt	0
G2S_cardedProgWonAmt	0.00
G2S_cardedWageredCashableAmt	0.00
G2S_cardedWageredNonCashAmt	0.00
G2S_cardedWageredPromoAmt	0.00
G2S_egmDispensedCashableAmt	0.00
G2S_egmDispensedNonCashAmt	0.00
G2S_egmDispensedPromoAmt	0.00
G2S_egmPaidBonusNonWonAmt	0.00
G2S_egmPaidBonusWonAmt	0.00
G2S_egmPaidGameWonAmt	0.00
G2S_egmPaidProgWonAmt	0.00
G2S_gamesSinceDoorClosedCnt	0
G2S_gamesSinceInitCnt	0
G2S_gamesSincePowerResetCnt	0
G2S_handPaidBonusNonWonAmt	0.00
G2S_handPaidBonusWonAmt	0.00
G2S_handPaidCancelAmt	0.00
G2S_handPaidGameWonAmt	0.00
G2S_handPaidProgWonAmt	0.00
G2S_logicDoorOpenCnt	0
G2S_playerCashableAmt	0.00
G2S_playerNonCashAmt	0.00
G2S_playerPromoAmt	0.00
G2S_wageredCashableAmt	0.00
G2S_wageredNonCashAmt	0.00
G2S_wageredPromoAmt	0.00

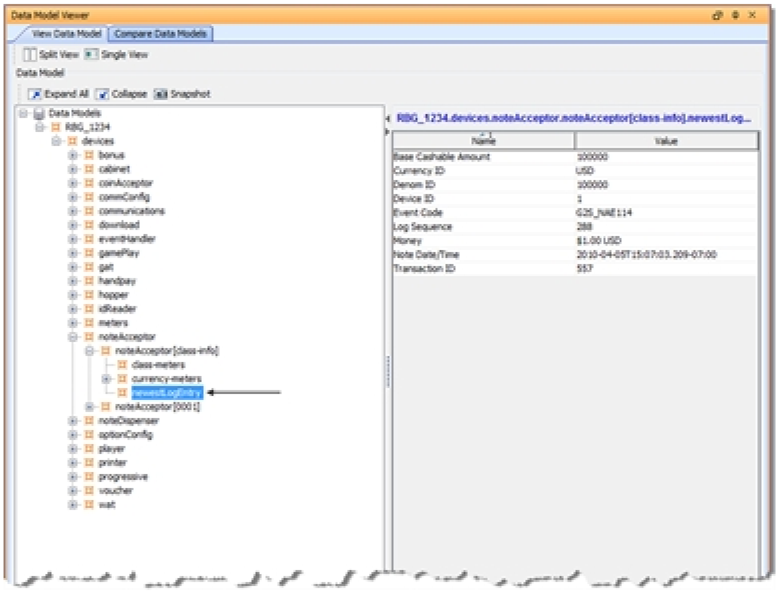
2. Click **Split View**.
3. Select the information you want to view for each data model view.

   If you want to view the data models for two different EGMs, select a second EGM ID in one of the views.

4. To return to a singular view, click **Single View**. The view on the left side of the screen is retained. The right-side panel view is retained for the next split view.

View Latest Log Entry for a Class

Logs are available for some G2S classes. You can view the last entry in the log by clicking the newestLogEntry entry under the class you want to view. Note that logs are not created until data for the log is created. For example, to see the note acceptor log (**Data Models** > {**EGM ID**} > **noteAcceptor** > **noteAcceptor [class-info]** > **newestLogEntry**) when none exists, insert a note into the EGM.



RBG_1234.devices.noteAcceptor.noteAcceptor[class-info].newestLog...

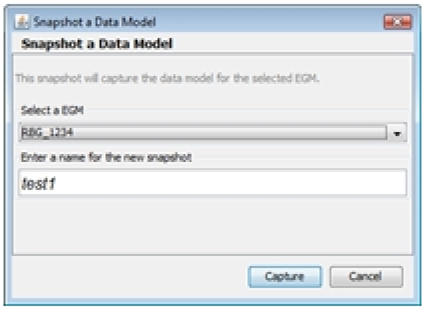
Name	Value
Base Cashable Amount	100000
Currency ID	USD
Denom ID	100000
Device ID	1
Event Code	G2S_NAE114
Log Sequence	288
Money	$1.00 USD
Note Date/Time	2010-04-05T15:07:03.209-07:00
Transaction ID	557

Snapshot a Data Model

The Snapshot option lets you take snapshots of one or all EGM data models as changes occur over the course of testing. Use this procedure to create a snapshot of the current EGM data model, for the selected EGM.

See About Snapshot Comparison for information on comparing data models.

1. Select the **View Data Model** tab from the Databases layout.
2. Click **Snapshot**.



Snapshot a Data Model

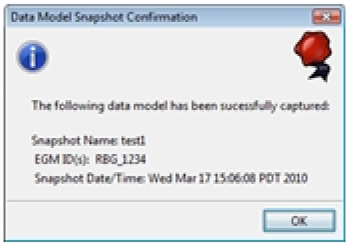
Snapshot a Data Model

This snapshot will capture the data model for the selected EGM.

Select a EGM: RBG_1234

Enter a name for the new snapshot: test1

3. Click the **Select an EGM** drop-down arrow, and select the EGM you want to snapshot or select All to snapshot all EGMs.
4. Type a name for the snapshot(s). If you are taking a snapshot of all EGMs, a snapshot with the defined name will be created for each EGM.
5. Click **Capture**.



Data Model Snapshot Confirmation

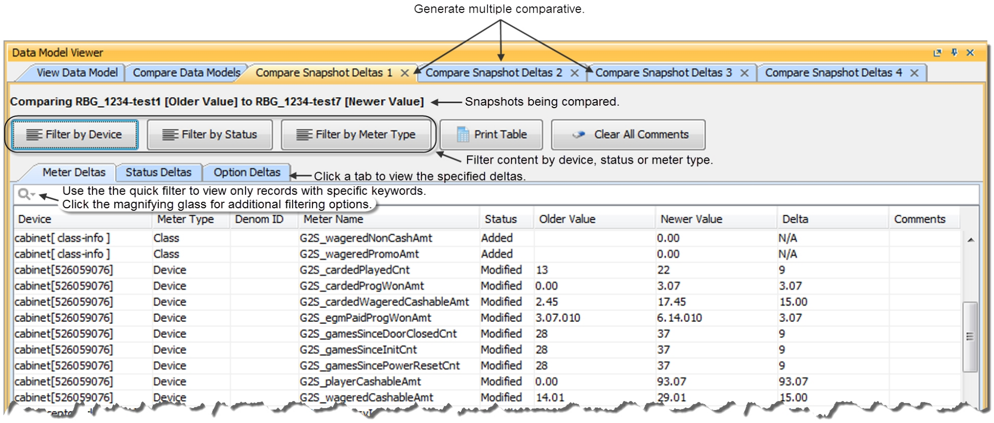
The following data model has been sucessfully captured:

Snapshot Name: test1
EGM ID(s): RBG_1234
Snapshot Date/Time: Wed Mar 17 15:06:08 PDT 2010

6. Review the snapshot summary information, and click **OK**.
7. Click the **Compare Data Models** tab to view the snapshots under the Available Snapshots list.

About Snapshot Comparison

When you compare two snapshots, the information is displayed in a separate Compare Snapshot Deltas tab and displayed by information type (options, meters and status). Device, status and meter type filters let you display just the information you want to see. In addition, you can dynamically filter data by typing keywords into the quick filter dialog at the top of each delta tab. You can add comments to the comparison output and print the current view of the active tab.



Generate multiple comparative.

Comparing RBG_1234-test1 [Older Value] to RBG_1234-test7 [Newer Value]

Snapshots being compared.

Filter content by device, status or meter type.

Click a tab to view the specified deltas.

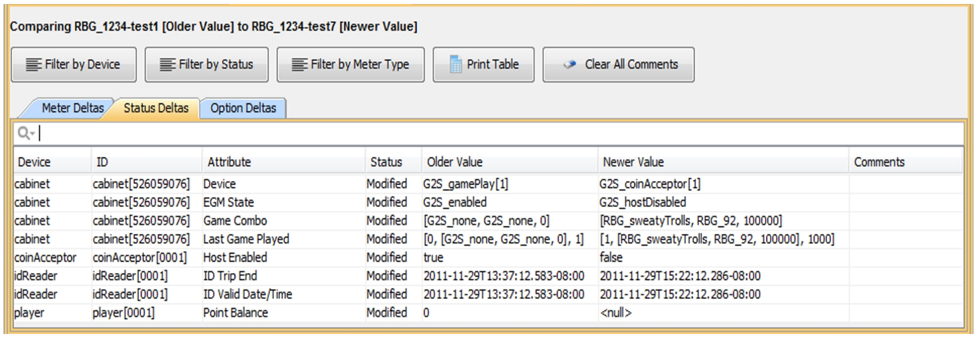
Use the the quick filter to view only records with specific keywords.
Click the magnifying glass for additional filtering options.

Device	Meter Type	Denom ID	Meter Name	Status	Older Value	Newer Value	Delta	Comments
cabinet[class-info]	Class		G2S_wageredNonCashAmt	Added		0.00	N/A	
cabinet[class-info]	Class		G2S_wageredPromoAmt	Added		0.00	N/A	
cabinet[526059076]	Device		G2S_cardedPlayedCnt	Modified	13	22	9	
cabinet[526059076]	Device		G2S_cardedProgWonAmt	Modified	0.00	3.07	3.07	
cabinet[526059076]	Device		G2S_cardedWageredCashableAmt	Modified	2.45	17.45	15.00	
cabinet[526059076]	Device		G2S_egmPaidProgWonAmt	Modified	3.07.010	6.14.010	3.07	
cabinet[526059076]	Device		G2S_gamesSinceDoorClosedCnt	Modified	28	37	9	
cabinet[526059076]	Device		G2S_gamesSinceInitCnt	Modified	28	37	9	
cabinet[526059076]	Device		G2S_gamesSincePowerResetCnt	Modified	28	37	9	
cabinet[526059076]	Device		G2S_playerCashableAmt	Modified	0.00	93.07	93.07	
cabinet[526059076]	Device		G2S_wageredCashableAmt	Modified	14.01	29.01	15.00	

Snapshot comparison report.

Each snapshot comparison generates three separate comparisons: Status Deltas, Meter Deltas and Option Deltas. The information displayed on each comparison includes: device name, device identifier, attribute name, comparison status, attribute value from the older snapshot, attribute value from the newer snapshot, and user-entered comments.

The Status Deltas comparison displays status parameters from the selected EGM snapshots.

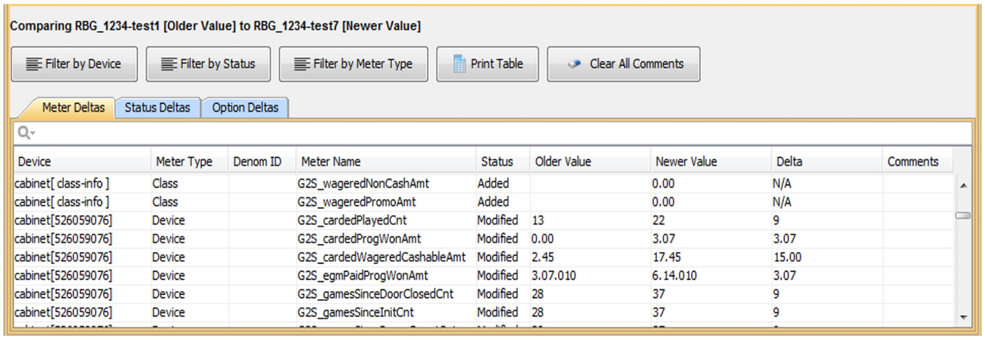


Comparing RBG_1234-test1 [Older Value] to RBG_1234-test7 [Newer Value]

Device	ID	Attribute	Status	Older Value	Newer Value	Comments
cabinet	cabinet[526059076]	Device	Modified	G2S_gamePlay[1]	G2S_coinAcceptor[1]	
cabinet	cabinet[526059076]	EGM State	Modified	G2S_enabled	G2S_hostDisabled	
cabinet	cabinet[526059076]	Game Combo	Modified	[G2S_none, G2S_none, 0]	[RBG_sweatyTrolls, RBG_92, 100000]	
cabinet	cabinet[526059076]	Last Game Played	Modified	[0, [G2S_none, G2S_none, 0], 1]	[1, [RBG_sweatyTrolls, RBG_92, 100000], 1000]	
coinAcceptor	coinAcceptor[0001]	Host Enabled	Modified	true	false	
idReader	idReader[0001]	ID Trip End	Modified	2011-11-29T13:37:12.583-08:00	2011-11-29T15:22:12.286-08:00	
idReader	idReader[0001]	ID Valid Date/Time	Modified	2011-11-29T13:37:12.583-08:00	2011-11-29T15:22:12.286-08:00	
player	player[0001]	Point Balance	Modified	0	<null>	

Status Deltas comparison.

The Meter Deltas comparison displays meter parameters from the selected EGM data models. The Meter Deltas comparison contains an additional Delta field that shows the meter difference between the older and newer snapshots. In the Delta field, amount differences are displayed with the appropriate decimals to make entries easier to read.

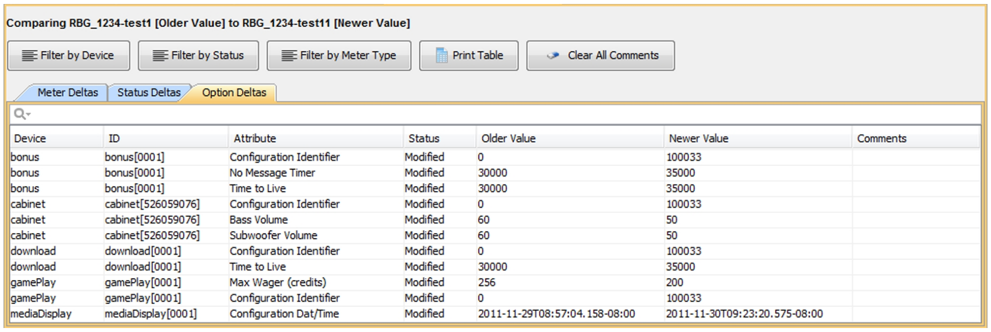


Comparing RBG_1234-test1 [Older Value] to RBG_1234-test7 [Newer Value]

Device	Meter Type	Denom ID	Meter Name	Status	Older Value	Newer Value	Delta	Comments
cabinet[class-info]	Class		G2S_wageredNonCashAmt	Added		0.00	N/A	
cabinet[class-info]	Class		G2S_wageredPromoAmt	Added		0.00	N/A	
cabinet[526059076]	Device		G2S_cardedPlayedCnt	Modified	13	22	9	
cabinet[526059076]	Device		G2S_cardedProgWonAmt	Modified	0.00	3.07	3.07	
cabinet[526059076]	Device		G2S_cardedWageredCashableAmt	Modified	2.45	17.45	15.00	
cabinet[526059076]	Device		G2S_egmPaidProgWonAmt	Modified	3.07.010	6.14.010	3.07	
cabinet[526059076]	Device		G2S_gamesSinceDoorClosedCnt	Modified	28	37	9	
cabinet[526059076]	Device		G2S_gamesSinceInitCnt	Modified	28	37	9	

Meter Deltas comparison.

The Option Deltas comparison displays option parameters from the selected EGM data models.



Comparing RBG_1234-test1 [Older Value] to RBG_1234-test11 [Newer Value]

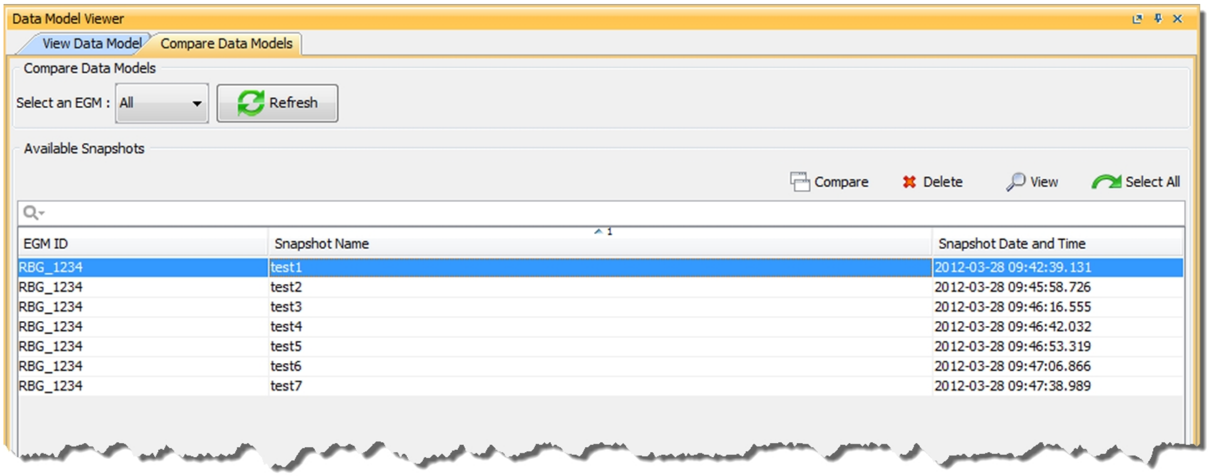
Device	ID	Attribute	Status	Older Value	Newer Value	Comments
bonus	bonus[0001]	Configuration Identifier	Modified	0	100033	
bonus	bonus[0001]	No Message Timer	Modified	30000	35000	
bonus	bonus[0001]	Time to Live	Modified	30000	35000	
cabinet	cabinet[526059076]	Configuration Identifier	Modified	0	100033	
cabinet	cabinet[526059076]	Bass Volume	Modified	60	50	
cabinet	cabinet[526059076]	Subwoofer Volume	Modified	60	50	
download	download[0001]	Configuration Identifier	Modified	0	100033	
download	download[0001]	Time to Live	Modified	30000	35000	
gamePlay	gamePlay[0001]	Max Wager (credits)	Modified	256	200	
gamePlay	gamePlay[0001]	Configuration Identifier	Modified	0	100033	
mediaDisplay	mediaDisplay[0001]	Configuration Dat/Time	Modified	2011-11-29T08:57:04.158-08:00	2011-11-30T09:23:20.575-08:00	

Option Deltas comparison.

Compare Snapshots

Use this procedure to view the differences between two EGM data model snapshots.

1. Select the **Compare Data Models** tab from the Data Model Viewer.



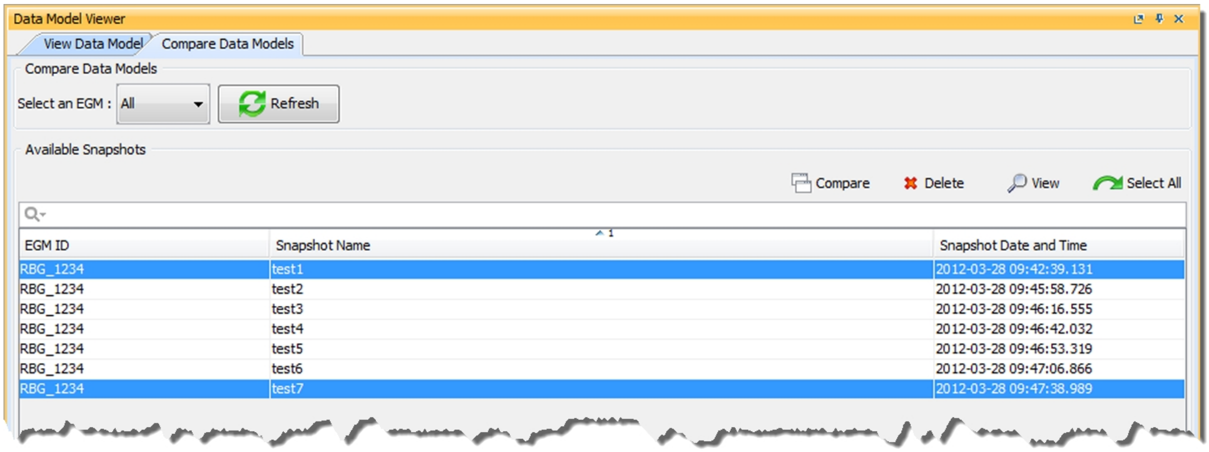
Select an EGM : All

EGM ID	Snapshot Name	Snapshot Date and Time
RBG_1234	test1	2012-03-28 09:42:39.131
RBG_1234	test2	2012-03-28 09:45:58.726
RBG_1234	test3	2012-03-28 09:46:16.555
RBG_1234	test4	2012-03-28 09:46:42.032
RBG_1234	test5	2012-03-28 09:46:53.319
RBG_1234	test6	2012-03-28 09:47:06.866
RBG_1234	test7	2012-03-28 09:47:38.989

2. Click **Refresh** to update the Available Snapshots list.

3. Click the **Select an EGM** drop-down to view only a specific EGM identifier (or leave the default **All** to display snapshots for all EGMs).

4. Hold down CTRL, and click the two snapshots you want to compare.



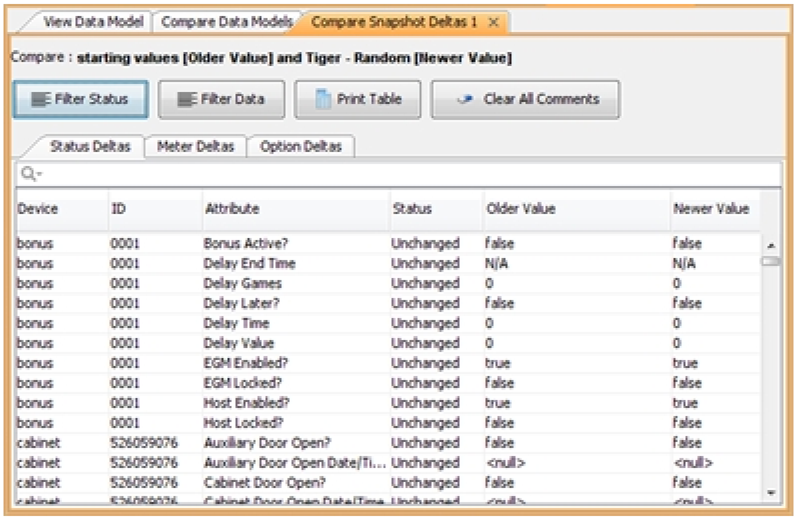
Select an EGM : All

EGM ID	Snapshot Name	Snapshot Date and Time
RBG_1234	test1	2012-03-28 09:42:39.131
RBG_1234	test2	2012-03-28 09:45:58.726
RBG_1234	test3	2012-03-28 09:46:16.555
RBG_1234	test4	2012-03-28 09:46:42.032
RBG_1234	test5	2012-03-28 09:46:53.319
RBG_1234	test6	2012-03-28 09:47:06.866
RBG_1234	test7	2012-03-28 09:47:38.989

3. Click **Compare** to generate the Compare Snapshot Deltas screen.

Filter Snapshot Comparison Data

There are three ways to filter comparison data on the Compare Snapshot screen: Filter Status, Filter Data and Quick Filter. You can use these options to filter out unwanted data.



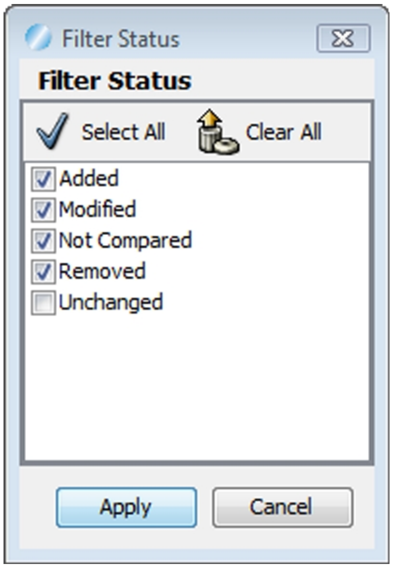
Compare : **starting values [Older Value] and Tiger - Random [Newer Value]**

Device	ID	Attribute	Status	Older Value	Newer Value
bonus	0001	Bonus Active?	Unchanged	false	false
bonus	0001	Delay End Time	Unchanged	N/A	N/A
bonus	0001	Delay Games	Unchanged	0	0
bonus	0001	Delay Later?	Unchanged	false	false
bonus	0001	Delay Time	Unchanged	0	0
bonus	0001	Delay Value	Unchanged	0	0
bonus	0001	EGM Enabled?	Unchanged	true	true
bonus	0001	EGM Locked?	Unchanged	false	false
bonus	0001	Host Enabled?	Unchanged	true	true
bonus	0001	Host Locked?	Unchanged	false	false
cabinet	526059076	Auxiliary Door Open?	Unchanged	false	false
cabinet	526059076	Auxiliary Door Open Date/Ti...	Unchanged	<null>	<null>
cabinet	526059076	Cabinet Door Open?	Unchanged	false	false
cabinet	526059076	Cabinet Door Open Date/Time	Unchanged	<null>	<null>

Filter Snapshot Comparison by Status

Use this option to filter information by comparison status.

1. Click **Filter Status**.



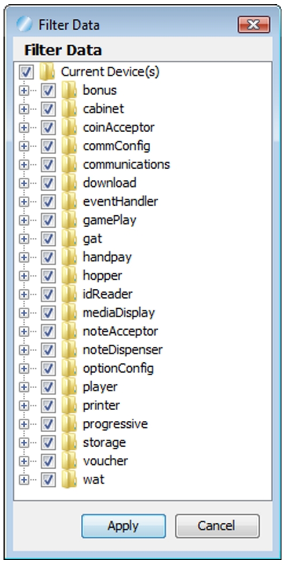
☑ Added
☑ Modified
☑ Not Compared
☑ Removed
□ Unchanged

2. Select or clear statuses as needed.
   - **Select All** - Click to select all statuses. Selected statuses are included in the comparison.
   - **Clear All** - Click to clear all statuses. Cleared statuses are filtered out of the comparison.
   - **Added** - Select to show device parameters with information that was added from the older snapshot to the newer snapshot.
   - **Modified** - Select to show device parameters with information that was modified from the older snapshot to the newer snapshot.
   - **Not Compared** - Select to show device parameters that were not compared. This status may indicate that an error occurred during comparison or that the values could not be compared.
   - **Removed** - Select to show device parameters that have been removed from the data model from the older snapshot to the newer snapshot.
   - **Unchanged** - Select to show device parameters with no changes from the older snapshot to the newer snapshot. By default, this status is cleared.
3. Click **Apply** to update the comparison view.

Filter Snapshot Comparison by Device

Use this option to filter information by device. *Only devices reported by the EGM are displayed*.

1. Click **Filter Data**.



Filter Data

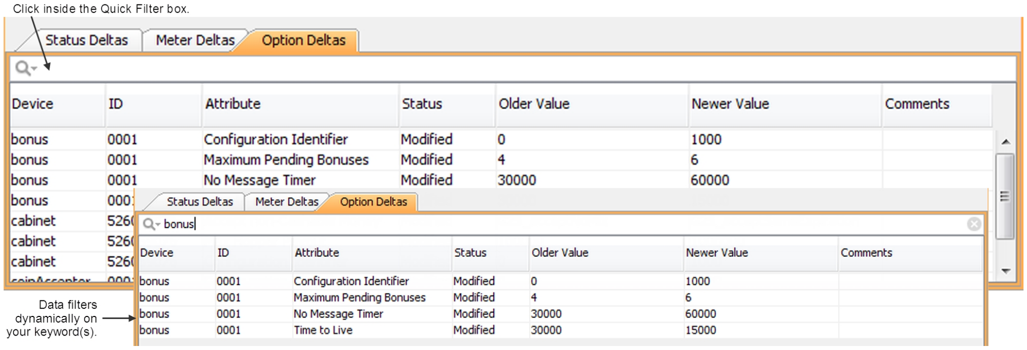
- ☑ Current Device(s)
  - ☑ bonus
  - ☑ cabinet
  - ☑ coinAcceptor
  - ☑ commConfig
  - ☑ communications
  - ☑ download
  - ☑ eventHandler
  - ☑ gamePlay
  - ☑ gat
  - ☑ handpay
  - ☑ hopper
  - ☑ idReader
  - ☑ mediaDisplay
  - ☑ noteAcceptor
  - ☑ noteDispenser
  - ☑ optionConfig
  - ☑ player
  - ☑ printer
  - ☑ progressive
  - ☑ storage
  - ☑ voucher
  - ☑ wat

2. Clear the checkbox of any devices you do not want to view.
3. Select the checkbox of any devices you want to view.
4. Click **Apply** to update the comparison.

Filter Snapshot Comparison View Quickly

Use this option to quickly filter the comparison data view by keyword(s).

1. Click inside the Quick Filter box, and type your keyword(s). As you type, only records containing the characters you type display.


Click inside the Quick Filter box.

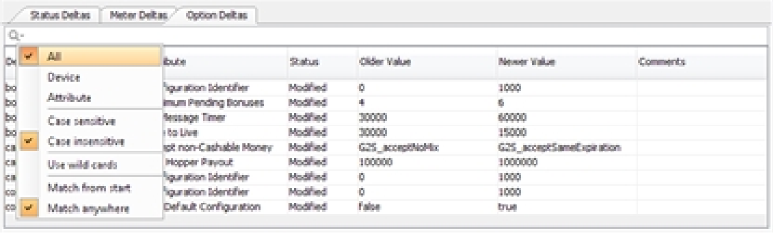
Device	ID	Attribute	Status	Older Value	Newer Value	Comments
bonus	0001	Configuration Identifier	Modified	0	1000	
bonus	0001	Maximum Pending Bonuses	Modified	4	6	
bonus	0001	No Message Timer	Modified	30000	60000	
bonus	000					
cabinet	526					
cabinet	526					
cabinet	526					

bonus

Device	ID	Attribute	Status	Older Value	Newer Value	Comments
bonus	0001	Configuration Identifier	Modified	0	1000	
bonus	0001	Maximum Pending Bonuses	Modified	4	6	
bonus	0001	No Message Timer	Modified	30000	60000	
bonus	0001	Time to Live	Modified	30000	15000	

Data filters dynamically on your keyword(s).

2. Click 🔍 to refine your filter.



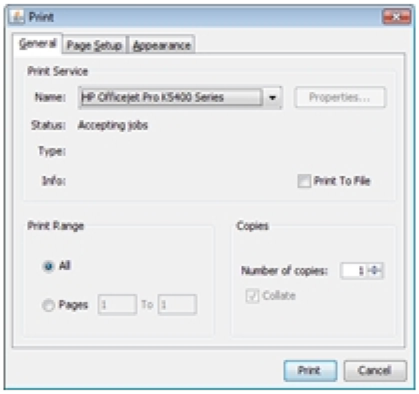
De	ibute	Status	Older Value	Newer Value	Comments
bo	figuration Identifier	Modified	0	1000	
bo	imum Pending Bonuses	Modified	4	6	
bo	Message Timer	Modified	30000	60000	
bo	to Live	Modified	30000	15000	
ca	ept non-Cashable Money	Modified	G2S_acceptNoMix	G2S_acceptSameExpiration	
ca	Hopper Payout	Modified	100000	1000000	
ca	figuration Identifier	Modified	0	1000	
co	figuration Identifier	Modified	0	1000	
co	Default Configuration	Modified	False	true	

3. Click on the filter options you want to use.
   - **All** - Select to filter on all columns. You can also select to sort on a specific column only. The column names you see depend on whether you are on the Compare Data Models tab or on a Compare Snapshot Deltas *n* tab.
   - **Case sensitive** - Select to consider title case in your filter entry.
   - **Case insensitive** - Select to disregard title case in your filter entry.
   - **Use wild cards** - Select to allow the use of wild cards in your filter entry. Acceptable wild cards are "?" (match any letter or digit) and "*" (match several letters or digits).
   - **Use regular expression** - Select to use Java regular expressions. For more information, go to: http://java.sun.com/developer/technicalArticles/releases/1.4regex/
   - **Match from start** - Select to filter from the start of your entry.
   - **Match exactly** - Select to filter on the exact character, spacing and case of the keyword(s).
   - **Match anywhere** - Select to filter on any part of your entry.
4. Click in the filter box, and type your keyword(s).

Print a Comparison Report

Use the Print Table option to print the currently displayed comparison to a report. Note that you can only print the active Deltas tab of the current Compare Snapshot Detail.

1. Right-click in the header of the comparison table you want to print, and select **Auto Resize All Columns**.
2. Click **Print Table**.



Name: HP Officejet Pro K5400 Series

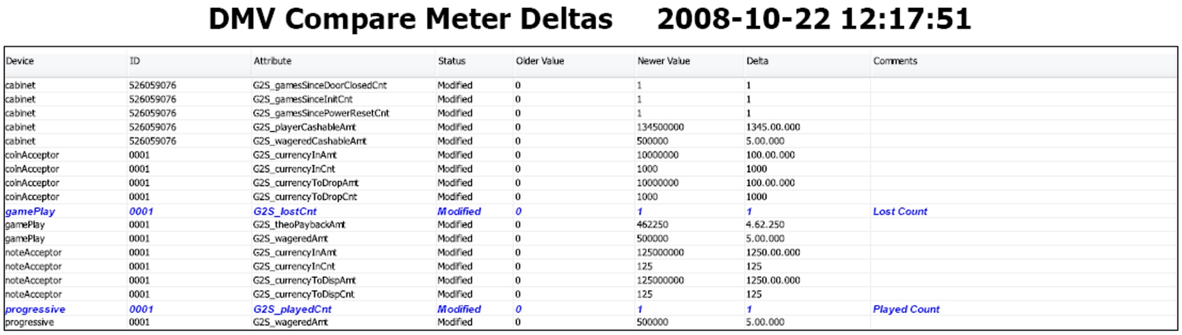
Status: Accepting jobs

● All

○ Pages 1 To 1

Number of copies: 1

3. Set the print properties as needed, and click **Print**.



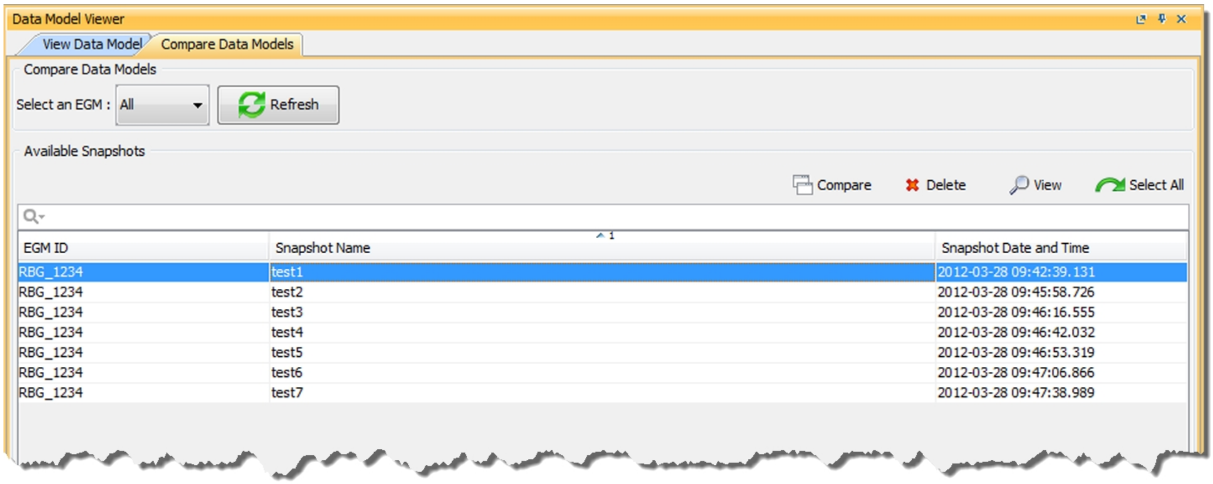
DMV Compare Meter Deltas 2008-10-22 12:17:51

Device	ID	Attribute	Status	Older Value	Newer Value	Delta	Comments
cabinet	526059076	G2S_gamesSinceDoorClosedCnt	Modified	0	1	1	
cabinet	526059076	G2S_gamesSinceInitCnt	Modified	0	1	1	
cabinet	526059076	G2S_gamesSincePowerResetCnt	Modified	0	1	1	
cabinet	526059076	G2S_playerCashableAmt	Modified	0	134500000	1345.00.000	
cabinet	526059076	G2S_wageredCashableAmt	Modified	0	500000	5.00.000	
coinAcceptor	0001	G2S_currencyInAmt	Modified	0	10000000	100.00.000	
coinAcceptor	0001	G2S_currencyInCnt	Modified	0	1000	1000	
coinAcceptor	0001	G2S_currencyToDropAmt	Modified	0	10000000	100.00.000	
coinAcceptor	0001	G2S_currencyToDropCnt	Modified	0	1000	1000	
gamePlay	***0001***	***G2S_lostCnt***	***Modified***	***0***	***1***	***1***	***Lost Count***
gamePlay	0001	G2S_theoPaybackAmt	Modified	0	462250	4.62.250	
gamePlay	0001	G2S_wageredAmt	Modified	0	500000	5.00.000	
noteAcceptor	0001	G2S_currencyInAmt	Modified	0	125000000	1250.00.000	
noteAcceptor	0001	G2S_currencyInCnt	Modified	0	125	125	
noteAcceptor	0001	G2S_currencyToDispAmt	Modified	0	125000000	1250.00.000	
noteAcceptor	0001	G2S_currencyToDispCnt	Modified	0	125	125	
progressive	***0001***	***G2S_playedCnt***	***Modified***	***0***	***1***	***1***	***Played Count***
progressive	0001	G2S_wageredAmt	Modified	0	500000	5.00.000	

Delete a Data Model Snapshot

Use the Delete Selected option on the Compare Data Models screen to delete one or more snapshots. Deleted snapshots are no longer available for comparison.

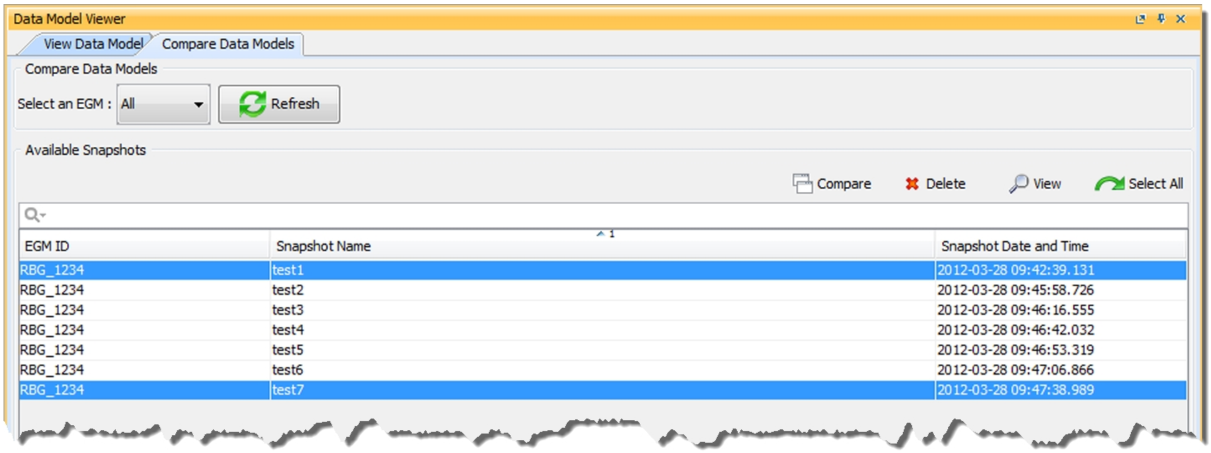
1. Select the **Compare Data Models** tab from the Data Model Viewer.



Select an EGM : All

EGM ID	Snapshot Name	Snapshot Date and Time
RBG_1234	test1	2012-03-28 09:42:39.131
RBG_1234	test2	2012-03-28 09:45:58.726
RBG_1234	test3	2012-03-28 09:46:16.555
RBG_1234	test4	2012-03-28 09:46:42.032
RBG_1234	test5	2012-03-28 09:46:53.319
RBG_1234	test6	2012-03-28 09:47:06.866
RBG_1234	test7	2012-03-28 09:47:38.989

2. Click the Select an EGM drop-down arrow, and choose to display all snapshots for a specific EGM or all EGMs.
3. Hold down **CTRL** and click the snapshots you want to delete, or click **Select All** to highlight all snapshots.



Select an EGM : All

EGM ID	Snapshot Name	Snapshot Date and Time
RBG_1234	test1	2012-03-28 09:42:39.131
RBG_1234	test2	2012-03-28 09:45:58.726
RBG_1234	test3	2012-03-28 09:46:16.555
RBG_1234	test4	2012-03-28 09:46:42.032
RBG_1234	test5	2012-03-28 09:46:53.319
RBG_1234	test6	2012-03-28 09:47:06.866
RBG_1234	test7	2012-03-28 09:47:38.989

4. Click **Delete Selected** to delete all selected snapshots.

Export an EGM Data Model

Use the Save As option to export an EGM data model to an XML file that you can then use with the RadBlue System Tester (RST) or the RadBlue Load Tester (RLT). For example, if:

- you don't have full-time access to a real EGM, you can save the data model from a real EGM and then load it into RST.
- you want RLT to mimic real EGMs, you can save multiple data models from real EGMs and load them into RLT.
- you have a real EGM, but it does not allow you the same access and control for testing as the RST. Simply save an EGM data model through RGS, and then load it into RST.

To capture and export a data model, RGS must be configured to communicate with a real EGM.

1. Go to: **Databases** > **Data Model Viewer**
2. Click **Save As**.
3. Navigate to the save location of your choice.
4. Change the default file name as required.
5. Click **Open**.

About Media Display

The mediaDisplay class allows host servers to display content on the EGM's Player User Interface (PUI). You can test mediaDisplay commands using RST, and view content through the Media Display tab on the SmartEGM layout.

> **Note:** The Media Display tab is disabled if there is no mediaDisplay defined in the SmartEGM configuration file.

The Media Display tab has two sliding panels - Media Display Content and Display.

From the Media Display Content panel you can view mediaDisplay content information for each mediaDisplay device, including content identifier, transaction identifier, log sequence number, current state and content location for each piece of loaded content.

From the Display panel, you can view the content being shown to the player. RST has two default media files to assist you in testing the mediaDisplay class. The default files are:

- **PPmain_test-2.swf** - A sample SWF file.
- **Asteroid_blaster.swf** - An asteroid blaster game.

If you are using RST with a G2S host system, you can download the sample SWF files from the following locations:

- **PPmain_test-2.swf** - http://www.radblue.com/downloads/flash/PPmain_test-2.swf
- **Asteroid_blaster.swf** - http://www.radblue.com/downloads/flash/Asteroid_blaster.swf

If you are using RGS with RST, these files are located in the [**install directory**] > **RGS webapps** > **RGS** > **MediaDisplayContent** folder. By default, the Media URI for the mediaDisplay.loadContent command is the asteroid blaster game from the RGS web server. However, you can change it, as needed, to the other RadBlue default file.

The following table shows how mediaDisplay commands effect RST.

When the host sends...	RST responds with...	And the Display panel shows....
loadContent	contentStatus	*Media Display Hidden*
setActiveContent	contentStatus	*Media Display Hidden*
showMediaDisplay	mediaDisplayAck	[Active Content]
hideMediaDisplay	mediaDisplayAck	*Media Display Hidden*
releaseContent	contentStatus	*Media Display Hidden*

Load and Display Content

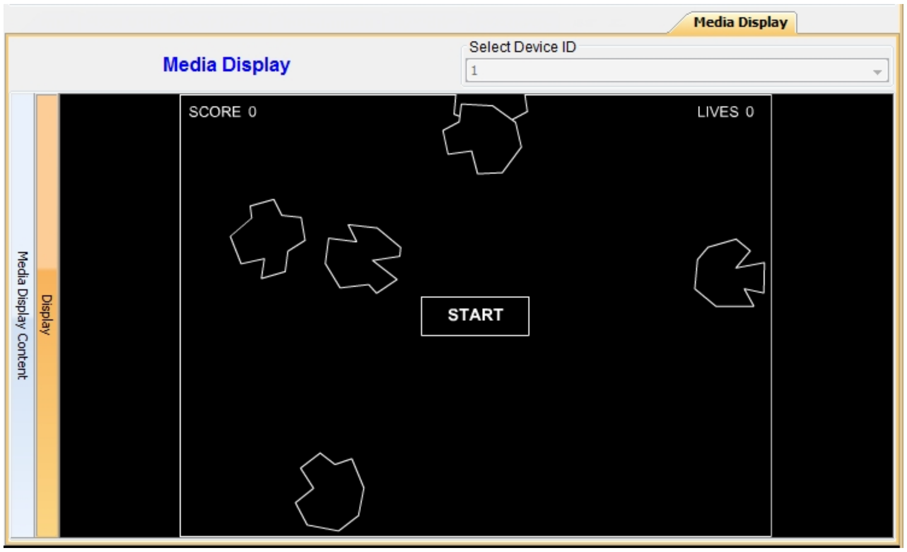
To use the `mediaDisplay` class in RST:

1. Go to the **SmartEGM** layout, and click **Start SmartEGM** to start the RST engine.
2. Click the **Media Display** tab on the SmartEGM layout.
3. Send the following commands from the G2S host system. If you are using RGS, send them from the Send Command layout.
    a. Define the **media URI**, and send `mediaDisplay.loadContent` to direct the EGM to load the specified content. The content information is displayed on the Media Display Content panel.
    b. Send `mediaDisplay.setActiveContent` to make the content active in RST.
    c. Send `mediaDisplay.showMediaDisplay` to run the content on the Display panel.
4. To remove a content file from the EGM, send `mediaDisplay.releaseContent`, being sure to define the content ID and transaction ID of the file you want to remove.

Play the Asteroid Blaster Game

The Asteroid Blaster game is one of three default RadBlue SWF files that you can use to test the `mediaDisplay` class. It is a working game that you can actually play.

1. Load and display the **Asteroid_blaster.swf** file.



2. Click **Start**.
3. **Play!**
   - Use the "up" arrow key to move your ship.
   - Use the "right" and "left" arrow keys to rotate the ship.
   - Press the space bar to fire.

About Installable Packages

To use installable packages, two commands must be sent from the host system:

1. `download.addPackage` - Adds a package to the EGM's package list.
2. `download.setScript` - Used to define the action that should be taken on the downloaded package (for example, install).

You must have an installable package to download before you can begin. Note that, because packages are EGM-specific, you can only use RadBlue packages with RST. The following:

- **Package 1** - http://www.radblue.com/downloads/smartegm/packages/package-1.zip
  This package contains two modules (RBG_module 1 and RGB_module 2). It can be installed, and modules are added to the module list.
- **Package 2** - http://www.radblue.com/downloads/smartegm/packages/package-2.zip
  This package contains two modules (RBG_module 3 and RGB_module 4). It can be installed, and modules are added to the module list.
- **Package 4** - http://www.radblue.com/downloads/smartegm/packages/package-4.zip
  This package contains the RBG_newGames module. When installed, four `gamePlay` devices are added to RST, which causes communications to cycle so a new descriptor list can be sent to all hosts.

Note that you can customize any RadBlue packages as needed for testing. See Installable Package Format for more information.

Once a package is installed, changes to the RST data model can be seen in the Data Model Viewer.

Installable Package Format

At the highest level, a RadBlue installable package is a ZIP file (WinZIP or Microsoft Zip format), containing the following:

1. An optional **readme.txt** file that describes the package. This file is not used in our tools.
2. A mandatory **package.xml** file that is used by RadBlue tools to provide information about the package (satisfying the G2S requests for information about the package). For more information, see PACKAGE.XML File Format.
3. One or more file folders, each of which can be considered a module within the package.

Each file folder contains one or more XML files, with a valid XML document of one or more gamePlay devices in valid **smartegm-config.xml** file format (validated against the **smartegm-config.xsd**). The package.xml file points at the XML files through its filename attribute so they can be installed.

Non-gamePlay devices can be present in the XML files, but are ignored (you can only install gamePlay devices).

The gamePlay device IDs within the XML files must be non-zero and unique.

RST Tests on Installable Packages

The following tests are performed by RST to determine if the package is an installable package:

1. Is it a valid ZIP file?
2. Is there a **package.xml** file in the root? (If so, it's a RadBlue package.)
3. Does the **package.xml** file list a package with **module-type= G2S_game**?
4. Modules are unpacked.
   a. Is the referenced file valid against our SmartEGM configuration schema?
   b. Are there gamePlay devices with unique, non-zero deviceId attributes.

If all of the above tests pass, the package can be installed.

PACKAGE.XML File Format

Name	Description
***Element*: package**	Contains data that is used to describe the package.
Attribute: description	This attribute describes the package. It is used in the G2S_packageContents command.
Attribute: release-number	A release number for the package. This attribute is also used in the packageContents command.
Attribute: namespace	This attribute declares the namespace for the XML document.
***Element*: module**	This element contains data that describes one module within this package. It can be repeated for multiple modules in one package.
Attribute: module-id	This attribute is used in moduleInfo in the packageContents command.
Attribute: description	This attribute is used in moduleInfo in the packageContents command.
Attribute: modlue-exception	This attribute is used for the moduleList.moduleStatus.modException command.
Attribute: module-type	This attribute is used in moduleInfo in the packageContents command. The module type for an installable package must be G2S_game.
Attribute: release-number	This attribute is used in moduleInfo in the packageContents command.
Attribute: status	This attribute is used for the moduleList.moduleStatus.modStatus command.
Attribute: filename	Information that points to the installable file within this package, for this module. This attribute is used when directed to install the package through a setScript command.
***Element*: storage**	
Attribute: device-class	These attributes are used to load the attributes in the storageUsed element of the moduleList.moduleStatus command when (and if) the modules in this package are installed.
Attribute: device-id	
Attribute: storage-id	
Attribute: storage-application	
Attribute: storage-used	
***Element*: algorithm**	
Attribute: algorithm-type	This attribute provides a list of the several authentication algorithms supported for this module.
End of Module Element	
***Element*: algorithm**	
Attribute: algorithm-type	This attribute provides a list of the several authentication algorithms supported for this package.
End of Module Element	

View Installable Package Data in RST

RST Data Model Updates

The RST Data Model Viewer allows you to see the current EGM data model. Any data model changes that are contained in a package can be viewed in the Data Model Viewer. Launch RST, select the **SmartEGM** layout, and select the **Data Model Viewer** sub-tab. Changes take effect after the RST restarts on a `download.setScript` command.

Message Transcript

View incoming `download` commands as well as outgoing responses in the Message Transcript. To view RST message traffic, select the **Transcript** layout. Then, click the **Transcript** object tab.

In the example above, installable package messages that are generated by RST after a `download.setScript` command are received.

Once authorization from all hosts have been received, RST automatically restarts in order to update its data model. This process is the same whether the `download.setScript` operation is to install or uninstall a package.

About Informed Player Extensions

Informed Player is a messaging extension of the G2S message protocol. The intent of the informed Player Class is to provide mechanism for enforcing Informed Player regulations at an EGM.

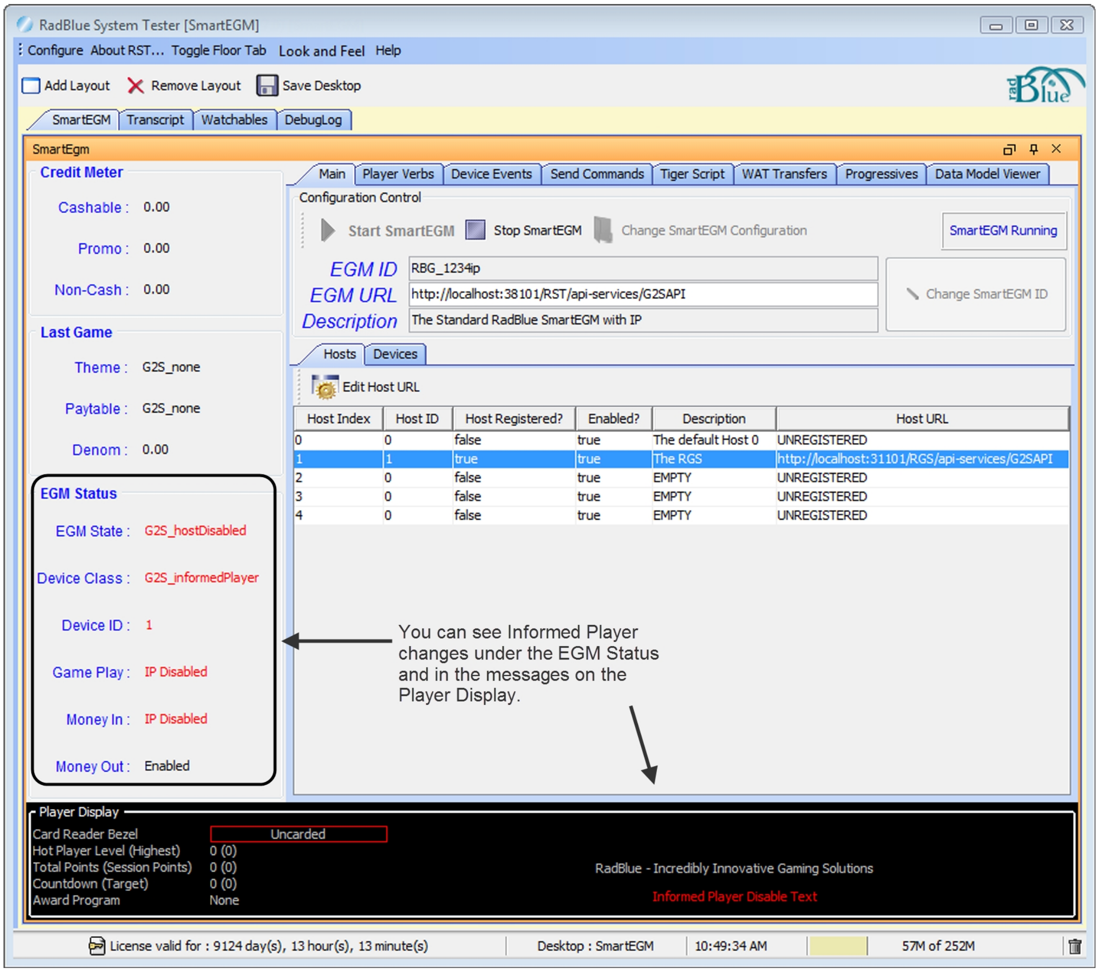
The Informed Player message protocol, *G2S Message Protocol Extension 1G1 Game To System*, is available on the GSA website.

Informed Player in RST

There are no options or functions for Informed Player in RST. However, RST supports all Informed Player messages and responds accordingly. You can see the result of Informed Player messages under the EGM Status section as well as in the player messages displayed in the Player Display.

Under EGM Status, there are three fields that display IP-specific information:

- **Device Class** - Indicates the device class that disabled the EGM.
- **Game Play** - Indicates whether game play is enabled at the EGM.
- **Money In** - Indicates whether the EGM is enabled to accept money.


RadBlue System Tester [SmartEGM]
Configure About RST... Toggle Floor Tab Look and Feel Help
Add Layout
Remove Layout
Save Desktop
SmartEGM Transcript Watchables DebugLog
SmartEgm
Main Player Verbs Device Events Send Commands Tiger Script WAT Transfers Progressives Data Model Viewer
Configuration Control
Start SmartEGM
Stop SmartEGM
Change SmartEGM Configuration
SmartEGM Running
Change SmartEGM ID
Hosts Devices
Edit Host URL

Credit Meter

Cashable : 0.00

Promo : 0.00

Non-Cash : 0.00

Last Game

Theme : G2S_none

Paytable : G2S_none

Denom : 0.00

EGM Status

EGM State : G2S_hostDisabled

Device Class : G2S_informedPlayer

Device ID : 1

Game Play : IP Disabled

Money In : IP Disabled

Money Out : Enabled

EGM ID: RBG_1234ip

EGM URL: http://localhost:38101/RST/api-services/G2SAPI

Description: The Standard RadBlue SmartEGM with IP

Host Index	Host ID	Host Registered?	Enabled?	Description	Host URL
0	0	false	true	The default Host 0	UNREGISTERED
1	1	true	true	The RGS	http://localhost:31101/RGS/api-services/G2SAPI
2	0	false	true	EMPTY	UNREGISTERED
3	0	false	true	EMPTY	UNREGISTERED
4	0	false	true	EMPTY	UNREGISTERED

You can see Informed Player changes under the EGM Status and in the messages on the Player Display.

Player Display

Card Reader Bezel	Uncarded
Hot Player Level (Highest)	0 (0)
Total Points (Session Points)	0 (0)
Countdown (Target)	0 (0)
Award Program	None

RadBlue - Incredibly Innovative Gaming Solutions

Informed Player Disable Text

License valid for : 9124 day(s), 13 hour(s), 13 minute(s) | Desktop : SmartEGM | 10:49:34 AM | 57M of 252M

To see messages as they are sent between RST and a G2S host, click the Transcript tab. You can use the Transcript to filter, sort and search through all commands sent and received by RST. Double-click any command to view its XML.

Configure RST for Informed Player (RST)

RST supports all of the `optionConfig` parameters documented in section 1.34.3.1 of the Informed Player message protocol. You can configure these parameter settings through the **smartegm-config-gsa-ip.xml** file.

How to Edit 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 Informed Player parameters as needed.
5. **Save** and **close** the file.
6. Launch RST.
   a. Go to the **RST Configuration Control** on the **Main** tab.
   b. Click **Change SmartEGM Configuration**.
   c. Click to select the configuration file you want to load, and click **Open**.

Get Going with Informed Player

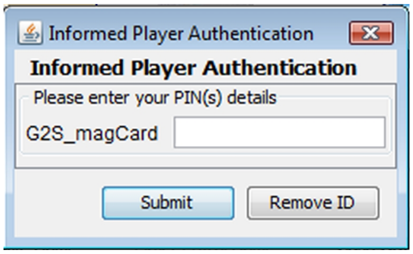
To get up and running with Informed Player, follow these steps.

1. Review Informed Player in RST
2. Configure RST for Informed Player
3. Start the SmartEGM Engine
4. Create EGM Activity
5. Send Informed Player Commands from the host
6. View Informed Player Activity

Create EGM Activity for Informed Player

You can create activity on RST from the SmartEGM tab. You can perform basic EGM operations from the Player Verbs tab, such as inserting a player card and playing a game.

If your Informed Player host implementation requires a player card and PIN, you are prompted to enter a password after you insert a player card:



Informed Player Authentication

Informed Player Authentication

Please enter your PIN(s) details

G2S_magCard

Type the password, and click **Save**.

About the RadBlue Smart Card Implementation

Since implementation of the smart card class requires the use of a proprietary Smart Card Application Protocol (SCAP), RadBlue has implemented the smart card class with its own version of a SCAP. This implementation demonstrates the smart card class between RST and RGS. Also, depending on your host or EGM implementation, there are three ways that you can use the RadBlue smart card:

1. RGS can be the smart card device owner if the EGM does not support SCAP.
2. RGS can be a guest of the smart card device if the EGM supports a proprietary SCAP implementation.*
3. If the host supports no SCAP, RST can be used as the EGM.

Your proprietary SCAP can be implemented in the tools as a custom project. Contact RadBlue for more information.

If you do not have RGS or RST, you can request a free student license. These two products communicate with each other by default, so set up and configuration is minimal.

*The Smart Card Editor in RGS (accessed through **Send Command** > **hostmsg**) can only be used when RGS is communicating with RST. Otherwise, when you send the `hostMsg` command, you are prompted for the ID number, hint and message data.

Smart Card in the Tester Toolkit

Smart card commands are available in Custom Scripting, the Startup Algorithm, the Response Manager and the Event Subscriptions component.

Track Changes to the Smart Card Device

The Data Model Viewer lets you view changes to the smart card device as well as compare smart card device options and meters at different points in a test.

Smart Card in the SmartEGM Configuration File

The SmartEGM configuration file in RST supports the smart card class. You can use the smartegm-config-smart-card.xml file that is installed with RST by default or you can add smart card information to a custom configuration file.

To change the SCAP used by RST, you must edit the **IGT_smartCardAppProtocol** value to the SCAP of your choice. The default value for the RadBlue SCAP implementation is **RBG_scap_v1**.

Finally, you can configure the ID reader device in the SmartEGM configuration file to self-validate an ID. If the `G2S_idValidMethod` parameter value for the ID reader device is G2S_self, the ID reader self-validates the ID and a `G2S_IDE101 Valid ID Presented` event is sent to the host. The SmartEGM validates the ID with the following information:

- **idFullName**: " "
- **idGender**: UNKNOWN
- **idLossLimit**: 0
- **idNumber**: card ID number
- **idPreferName**: " "
- **idValidSource**: SELF
- **localeId**: cabinetProfile.localId
- **playerId**: card ID number

Get RST and RGS Talking

1. Install RGS and RST on the same computer.
2. Launch both products from their respective desktop icons.
3. In RST, go to the SmartEGM layout, and click **Change SmartEGM Configuration**.
4. Select **smartegm-config-smart-card**, and click **Open**.
5. Click the **Devices** subtab on the Main tab, and verify that the **IGT_smartcard** device is present.
6. Click **Start SmartEGM** in the SmartEGM Control section at the top of the layout.
7. Go to RGS, click the **Engine** layout, and verify that the **RBG** EGM is listed under **Connected EGM(s)**. This means that RGS and RST have established communications.

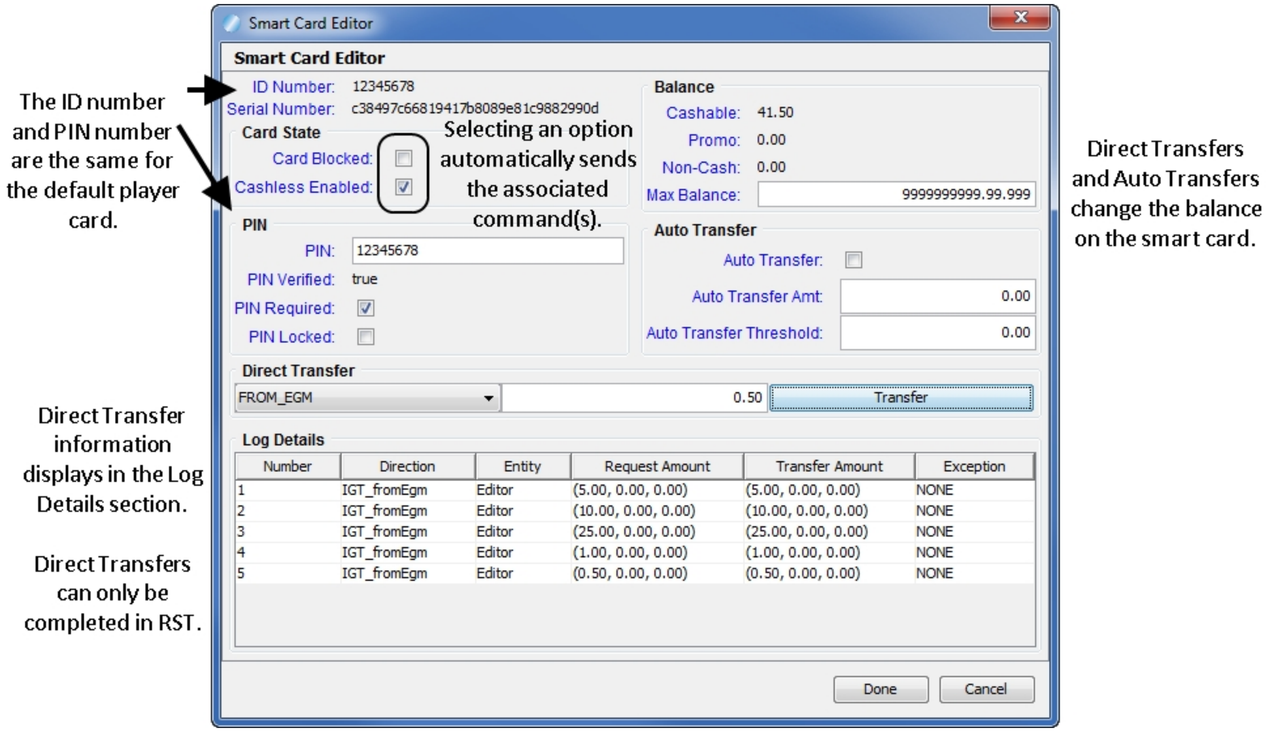
Insert the Smart Card

1. In RST, go to the **SmartEGM** layout, and select the **Player Verbs** tab.
2. Click the **Smart Card** bar to view smart card options.
3. Click **Insert ID**.
4. Type ID Number **12345678**, and click **Insert ID**.
5. Look at the Message Transcript in either RST or RGS. *RST generates* `IGT_SCE101 Smart Card Validated`.
6. Click **Enter PIN**.
7. Type Auth Code **12345678**, and click **Enter PIN**. By default, the PIN is the same as the card ID number.
   - *RST generates* `IGT_SCE104 Smart Card PIN Verified`.

How to Use the Smart Card Editor

The Smart Card Editor options let you generate smart card commands from either RST or RGS. In RST, the Smart Card Editor can be accessed from the **Smart Card** bar on the **Player Verbs** tab. In RGS, the Smart Card Editor can be accessed by going to **Send Command** > **IGT_smartCard** > **hostMsg**.

Note that the RGS Smart Card Editor can be used only when RGS is communicating with RST. If RST (or the EGM) is not using the default RadBlue SCAP, you do not see the RGS Smart Card Editor when you select the `hostMsg` command. Instead, you are prompted for the ID number, hint and message data.



The ID number and PIN number are the same for the default player card.

Selecting an option automatically sends the associated command(s).

Direct Transfers and Auto Transfers change the balance on the smart card.

Direct Transfer information displays in the Log Details section.

Direct Transfers can only be completed in RST.

ID Number: 12345678
Serial Number: c38497c66819417b8089e81c9882990d

Card State
Card Blocked: □
Cashless Enabled: ☑

PIN
PIN: 12345678
PIN Verified: true
PIN Required: ☑
PIN Locked: □

Balance
Cashable: 41.50
Promo: 0.00
Non-Cash: 0.00
Max Balance: 9999999999.99.999

Auto Transfer
Auto Transfer: □
Auto Transfer Amt: 0.00
Auto Transfer Threshold: 0.00

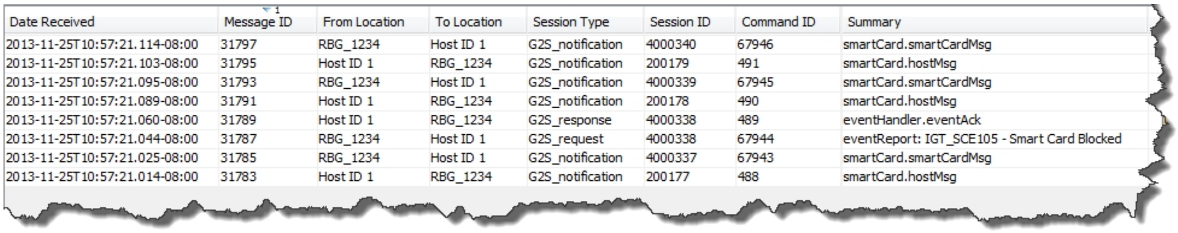
Direct Transfer
FROM_EGM 0.50

Log Details

Number	Direction	Entity	Request Amount	Transfer Amount	Exception
1	IGT_fromEgm	Editor	(5.00, 0.00, 0.00)	(5.00, 0.00, 0.00)	NONE
2	IGT_fromEgm	Editor	(10.00, 0.00, 0.00)	(10.00, 0.00, 0.00)	NONE
3	IGT_fromEgm	Editor	(25.00, 0.00, 0.00)	(25.00, 0.00, 0.00)	NONE
4	IGT_fromEgm	Editor	(1.00, 0.00, 0.00)	(1.00, 0.00, 0.00)	NONE
5	IGT_fromEgm	Editor	(0.50, 0.00, 0.00)	(0.50, 0.00, 0.00)	NONE

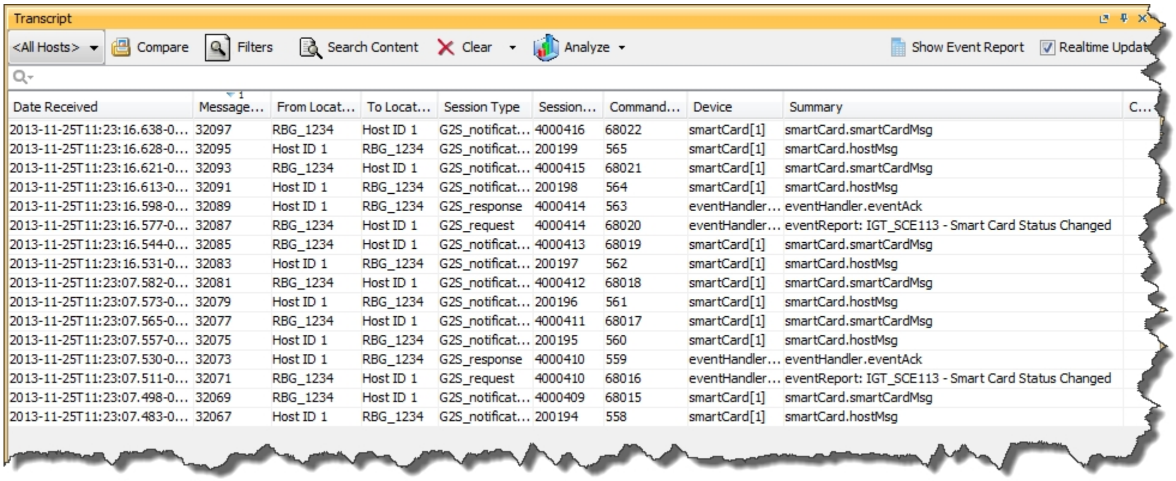
Note the following when working with the editor:

1. **Actions on the Smart Card Editor panel are dynamic**.
   There is no need to press **OK** to complete an action. For example, if you select **Card Blocked**, you immediately see the following commands in the Message Transcript:



Date Received	Message ID	From Location	To Location	Session Type	Session ID	Command ID	Summary
2013-11-25T10:57:21.114-08:00	31797	RBG_1234	Host ID 1	G2S_notification	4000340	67946	smartCard.smartCardMsg
2013-11-25T10:57:21.103-08:00	31795	Host ID 1	RBG_1234	G2S_notification	200179	491	smartCard.hostMsg
2013-11-25T10:57:21.095-08:00	31793	RBG_1234	Host ID 1	G2S_notification	4000339	67945	smartCard.smartCardMsg
2013-11-25T10:57:21.089-08:00	31791	Host ID 1	RBG_1234	G2S_notification	200178	490	smartCard.hostMsg
2013-11-25T10:57:21.060-08:00	31789	Host ID 1	RBG_1234	G2S_response	4000338	489	eventHandler.eventAck
2013-11-25T10:57:21.044-08:00	31787	RBG_1234	Host ID 1	G2S_request	4000338	67944	eventReport: IGT_SCE105 - Smart Card Blocked
2013-11-25T10:57:21.025-08:00	31785	RBG_1234	Host ID 1	G2S_notification	4000337	67943	smartCard.smartCardMsg
2013-11-25T10:57:21.014-08:00	31783	Host ID 1	RBG_1234	G2S_notification	200177	488	smartCard.hostMsg

2. **You *must* press "Enter" after you type information into some text input fields**.
   This includes the PIN, Auto Transfer Amt and Auto Transfer Threshold fields. For example, from the RGS Smart Card Editor, select **Auto Transfer**. Then, type **20.00** in the **Auto Transfer Amt** field, and press **Enter**. You see the following commands in the Message Transcript:



Date Received	Message...	From Locat...	To Locat...	Session Type	Session...	Command...	Device	Summary	C...
2013-11-25T11:23:16.638-0...	32097	RBG_1234	Host ID 1	G2S_notificat...	4000416	68022	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:16.628-0...	32095	Host ID 1	RBG_1234	G2S_notificat...	200199	565	smartCard[1]	smartCard.hostMsg	
2013-11-25T11:23:16.621-0...	32093	RBG_1234	Host ID 1	G2S_notificat...	4000415	68021	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:16.613-0...	32091	Host ID 1	RBG_1234	G2S_notificat...	200198	564	smartCard[1]	smartCard.hostMsg	
2013-11-25T11:23:16.598-0...	32089	Host ID 1	RBG_1234	G2S_response	4000414	563	eventHandler...	eventHandler.eventAck	
2013-11-25T11:23:16.577-0...	32087	RBG_1234	Host ID 1	G2S_request	4000414	68020	eventHandler...	eventReport: IGT_SCE113 - Smart Card Status Changed	
2013-11-25T11:23:16.544-0...	32085	RBG_1234	Host ID 1	G2S_notificat...	4000413	68019	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:16.531-0...	32083	Host ID 1	RBG_1234	G2S_notificat...	200197	562	smartCard[1]	smartCard.hostMsg	
2013-11-25T11:23:07.582-0...	32081	RBG_1234	Host ID 1	G2S_notificat...	4000412	68018	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:07.573-0...	32079	Host ID 1	RBG_1234	G2S_notificat...	200196	561	smartCard[1]	smartCard.hostMsg	
2013-11-25T11:23:07.565-0...	32077	RBG_1234	Host ID 1	G2S_notificat...	4000411	68017	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:07.557-0...	32075	Host ID 1	RBG_1234	G2S_notificat...	200195	560	smartCard[1]	smartCard.hostMsg	
2013-11-25T11:23:07.530-0...	32073	Host ID 1	RBG_1234	G2S_response	4000410	559	eventHandler...	eventHandler.eventAck	
2013-11-25T11:23:07.511-0...	32071	RBG_1234	Host ID 1	G2S_request	4000410	68016	eventHandler...	eventReport: IGT_SCE113 - Smart Card Status Changed	
2013-11-25T11:23:07.498-0...	32069	RBG_1234	Host ID 1	G2S_notificat...	4000409	68015	smartCard[1]	smartCard.smartCardMsg	
2013-11-25T11:23:07.483-0...	32067	Host ID 1	RBG_1234	G2S_notificat...	200194	558	smartCard[1]	smartCard.hostMsg	

   Notice that the Smart Card Editor in RST now reflects an **Auto Transfer Amt** of **20.00**.

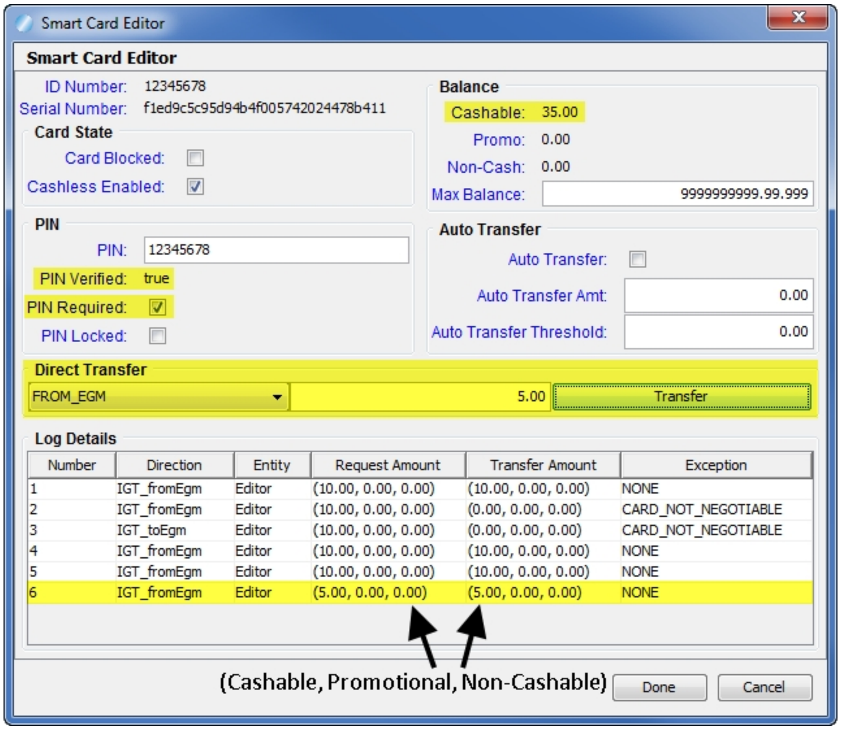
3. **After you change the PIN, remove the smart card and re-insert it to use the new PIN**.
4. **If a PIN is required, the "PIN Verified" field must be "true" to transfer funds**.
5. **The Log Details section provides information on direct transfers *only***.

Pre-Load Funds onto a Smart Card

You can pre-load funds onto the smart card through the Smart Card Editor in RST.

> **Note:** While the fields described below are also available on the RGS Smart Card Editor, they are not functioning. **You cannot perform a direct transfer from RGS**.

1. In RST, go to **SmartEGM** > **Player Verbs** > **Smart Card** > **Modify Smart Card**.
2. If **PIN Required** is selected, verify that the **PIN Verified** value is **true**.
3. Click the **Direct Transfer** drop-down, and select **FROM_EGM**.
4. Type the dollar amount to be transferred onto the smart card. For this example, we typed **5.00**.
5. Click **Transfer**.



Smart Card Editor

ID Number: 12345678
Serial Number: f1ed9c5c95d94b4f005742024478b411

Card State
Card Blocked: □
Cashless Enabled: ☑

Balance
Cashable: 35.00
Promo: 0.00
Non-Cash: 0.00
Max Balance: 9999999999.99.999

PIN
PIN: 12345678
PIN Verified: true
PIN Required: ☑
PIN Locked: □

Auto Transfer
Auto Transfer: □
Auto Transfer Amt: 0.00
Auto Transfer Threshold: 0.00

Direct Transfer
FROM_EGM 5.00

Log Details

Number	Direction	Entity	Request Amount	Transfer Amount	Exception
1	IGT_fromEgm	Editor	(10.00, 0.00, 0.00)	(10.00, 0.00, 0.00)	NONE
2	IGT_fromEgm	Editor	(10.00, 0.00, 0.00)	(0.00, 0.00, 0.00)	CARD_NOT_NEGOTIABLE
3	IGT_toEgm	Editor	(10.00, 0.00, 0.00)	(0.00, 0.00, 0.00)	CARD_NOT_NEGOTIABLE
4	IGT_fromEgm	Editor	(10.00, 0.00, 0.00)	(10.00, 0.00, 0.00)	NONE
5	IGT_fromEgm	Editor	(10.00, 0.00, 0.00)	(10.00, 0.00, 0.00)	NONE
6	IGT_fromEgm	Editor	(5.00, 0.00, 0.00)	(5.00, 0.00, 0.00)	NONE

The amount transferred (in this case, 5.00) is added to the smart card balance and the transfer is recorded in the Log Details. Note that if a PIN is required, but has not been authenticated, a **Card Not Negotiable** error is logged and the transfer does not occur.

Auto-Transfer Smart Card Funds

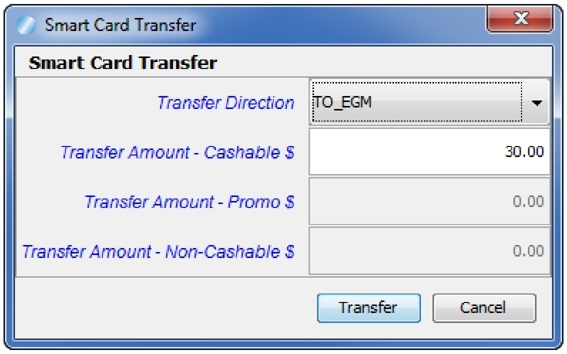
Generate the commands for an auto-transfer of funds by setting the amount and threshold through either RGS or RST. Then, simulate game play in RST.

1. In RST, go to **SmartEGM** > **Player Verbs** > **Smart Card**.
2. Insert the smart card.
3. Add money to the RST credit meter.
   a. Click **Insert Note(s)**.
   b. Verify that the **Note Count** is **1**.
   c. Click the **Note To Insert** drop-down, and select **$10.00 USD**.
   d. Notice that the Credit Meter's **Cashable** field now displays **10.00**.
4. Click **Modify Smart Card**.
5. Select **Auto Transfer** to enable the auto-transfer functionality.
6. Type the amount that you want to transfer (**Auto Transfer Amt**). For this example, we typed **10.00**.
7. Type the credit meter amount that triggers the auto-transfer (**Auto Transfer Threshold**). For this example, we typed **10.00**.
8. Click **Done** to close the editor.
9. Click **Play Simple Game**.
10. Set the following fields:
    - **Denom ID** - $1.00 USD
    - **Cashable Wager** - 2
    - **Primary Win** - 1
11. Click **Play Game**.
12. Look at the Credit Meter's Cashable field. It should increment to **19.00** if the auto-transfer was successful.
13. Look at the Message Transcript to see the smart card commands for auto-transfers.

Transfer Smart Card Funds

The Transfer Funds option lets you move funds from the smart card to the EGM or from the EGM to the smart card.

1. In RST, go to the **SmartEGM** layout, and select the **Player Verbs** tab.
2. Click the **Smart Card** bar to view smart card options.
3. Add $10.00 to the RST credit meter.
4. Add $30.00 to the smart card.
4. Click **Transfer Funds**.



Smart Card Transfer

Smart Card Transfer

Transfer Direction: TO_EGM

Transfer Amount - Cashable $: 30.00

Transfer Amount - Promo $: 0.00

Transfer Amount - Non-Cashable $: 0.00

5. Click the **Transfer Direction** drop-down, and select to transfer smart card funds to the EGM (**TO EGM**) or to transfer EGM funds to the smart card (**FROM EGM**). For this example, we chose **TO EGM**.
6. The **Transfer Amount - Cashable $** is the dollar amount of cashable funds available on either the smart card (if you select TO EGM) or the EGM (if you select FROM EGM). We chose **TO EGM**, so this field reflects the **Current Balance** of the smart card.
7. Click **Transfer**. The **Credit Meter** now displays a **Cashable** amount of **40.00**.

Add a Player Card with a Smart Card ID Reader Type

You can add a player card through the RGS Player Database with a `G2S_smartcard` ID reader type. Note that once you save the new player card record, you **cannot** edit the ID Reader Type value.

To add a player card with a Smart Card ID reader type:

1. In RGS, go to **Databases** > **Player Database**.
2. Click **Add Player**.
3. Click **ID Reader Type** the drop-down, and select **G2S_smartcard**.
4. Configure the player card as needed.
5. Click **Save**.

Run a Tiger Script with Smart Card

A Tiger script that uses smart card features is installed with RST. You can modify this script, if needed, but the default script does the following:

1. Inserts an ID (a smart card).
2. Inserts a $1 note.
3. If the smart card is not negotiable, enters the smart card's PIN.
4. Transfers the player funds from the EGM to the smart card.
5. Removes the ID.

To load and run the smart card Tiger script:

1. In RST, select the **SmartEGM** layout.
2. Click the **Tiger Script** tab.
3. Click **Select Script**.
4. Select **smartegm-example-smart-card-001**, and click **Open**. The script loads into the script runner.
5. Click **Start Script**.

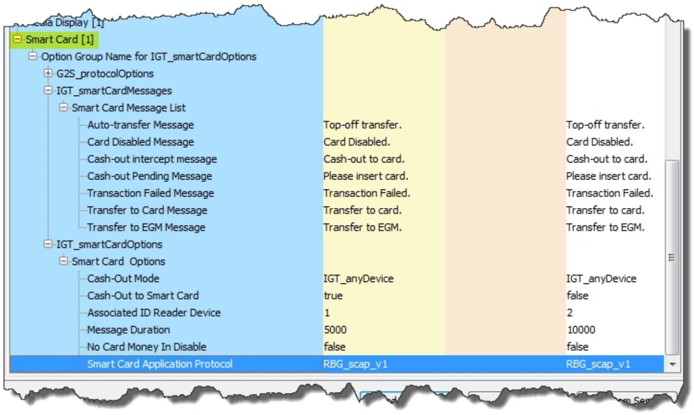
   When the script completes, status information is displayed.

Change Smart Card Configuration Options

Edit smart card configuration options through the `optionConfig.setOptionChange` command in RGS. Note that some options, including the Smart Card Application Protocol option, are read-only. If RGS is communicating with RST, you can change the SCAP through the SmartEGM configuration file.

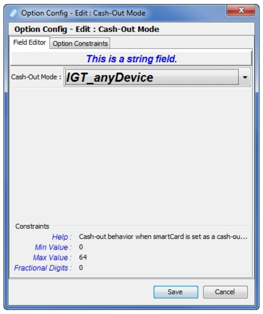
To modify smart card configuration options:

1. Go to the **Send Command** layout.
2. Select **G2S_optionConfig** from the Current Devices list, and double click the **setOptionChange** command.
3. Scroll to the **Smart Card** device you want to modify, and click the plus icon (+) to expand it.
4. Continue to drill down to the smart card configuration option(s) you want to change.



...dia Display [1]			
⊟ Smart Card [1]			
⊟ Option Group Name for IGT_smartCardOptions			
⊞ G2S_protocolOptions			
⊟ IGT_smartCardMessages			
⊟ Smart Card Message List			
Auto-transfer Message	Top-off transfer.		Top-off transfer.
Card Disabled Message	Card Disabled.		Card Disabled.
Cash-out intercept message	Cash-out to card.		Cash-out to card.
Cash-out Pending Message	Please insert card.		Please insert card.
Transaction Failed Message	Transaction Failed.		Transaction Failed.
Transfer to Card Message	Transfer to card.		Transfer to card.
Transfer to EGM Message	Transfer to EGM.		Transfer to EGM.
⊟ IGT_smartCardOptions			
⊟ Smart Card Options			
Cash-Out Mode	IGT_anyDevice		IGT_anyDevice
Cash-Out to Smart Card	true		false
Associated ID Reader Device	1		2
Message Duration	5000		10000
No Card Money In Disable	false		false
Smart Card Application Protocol	RBG_scap_v1		RBG_scap_v1

5. Double-click an option to edit it. For this example, we selected **Cash-Out Mode** under **Option Group Name for IGT_smartCardOptions** > **IGT_smartCardOptions** > **Smart Card Options**.



Option Config - Edit : Cash-Out Mode

This is a string field.

Cash-Out Mode : *IGT_anyDevice*

Constraints

Help : Cash-out behavior when smartCard is set as a cash-ou...
Min Value : 0
Max Value : 64
Fractional Digits : 0

6. Click the **Cash-Out Mode** drop-down, and select **IGT_smartCardOnly**.
7. Click **Save**.
8. Click **Send Command**.

Send Smart Card Commands from RGS

You can send smart card commands from the RGS Send Command layout. The Message Transcript, located at the bottom of the Send Command layout, displays each message and the response from the RST.

The following smart card commands are available in RGS:

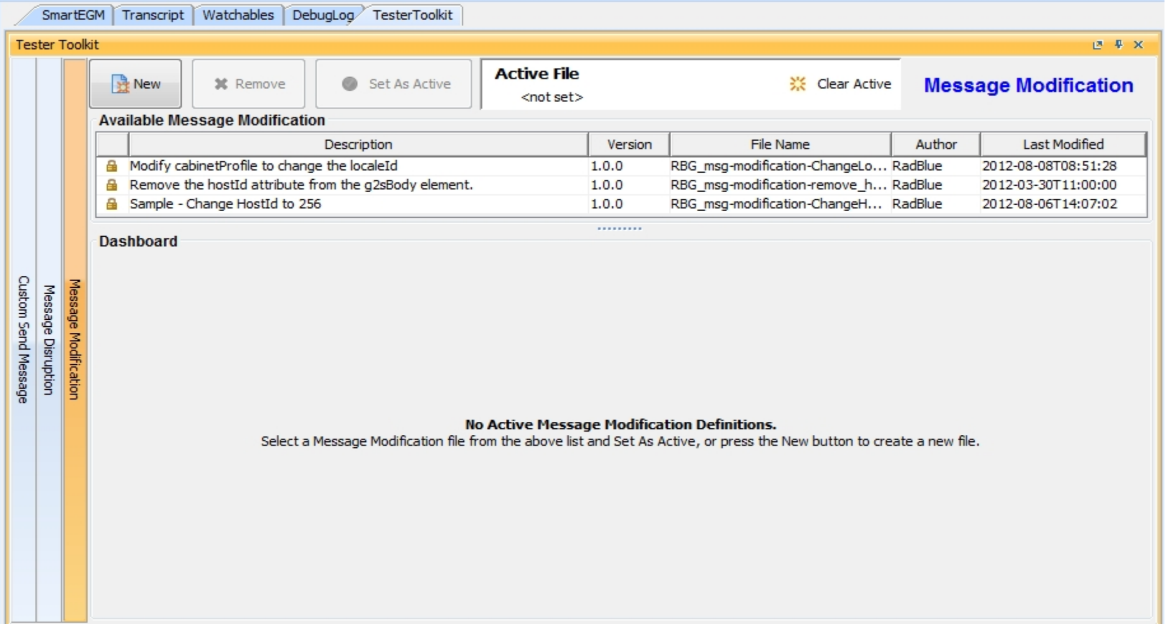
- `getSmartCardLog`
- `getSmartCardLogStatus`
- `getSmartCardProfile`
- `getSmartCardStatus`
- `hostMsg`
- `setSmartCardLockOut`
- `setSmartCardState`

See the G2S Protocol document for additional information on each of these commands.

About the Tester Toolkit

The Tester Toolkit is an optional module for RST. This module allows you to extend normal testing by customizing messages sent by the RST. Using the sidebars, you can:

- Create a list of modifications that RST makes to specified outgoing messages, using the Message Modification panel.
- Create a new definition or select a message disruption definition from the available list to edit, using the Message Disruption panel.
- Send messages that you create to test: the SOAP layer, whether the host system can handle missing required values, or illegal values by generating the correct error in response, using the Custom Send Message panel.



Active File
<not set>

Available Message Modification

	Description	Version	File Name	Author	Last Modified
🔒	Modify cabinetProfile to change the localeId	1.0.0	RBG_msg-modification-ChangeLo...	RadBlue	2012-08-08T08:51:28
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0	RBG_msg-modification-remove_h...	RadBlue	2012-03-30T11:00:00
🔒	Sample - Change HostId to 256	1.0.0	RBG_msg-modification-ChangeH...	RadBlue	2012-08-06T14:07:02

Dashboard

No Active Message Modification Definitions.
Select a Message Modification file from the above list and Set As Active, or press the New button to create a new file.

Send Custom Messages

About Custom Messages

You can use the Send a Custom Message functionality in the RST Tester Toolkit to test the SOAP layer, as well as whether the host system can handle missing required values or illegal values, by verifying that the host generated the correct error in response. To do this, you can change the following elements of any G2S message:

- **EGM ID for SOAP** – EGM identifier to put in the SOAP message. If there is no value, the current EGM ID is used. This is normally a fixed value.
- **Fix Up** – Indicates whether RST should add valid XML around a custom message or send it exactly as entered.
- **Host ID** – Identifier of host system to which you are sending the message.
- **Host ID for SOAP** – Host identifier to put in the SOAP message. If there is no value, use the current host ID. This is normally a fixed value.
- **Message XML** – Message XML that is changed if the Fix Up value is **true**.

Send a Custom Message

The custom Message option lets you send custom messages that can be "fixed up" with auto-generated content as needed. To use this option:

- Enter, or paste from the transcript, an entire G2S message into the XML Message text box (required).
- Delete the command ID or other attributes you would like the RST to supply.
- Make sure the device class and the device ID are still in the message (required).
- Delete any of the fix up attributes as needed.
- Insert a "-1" value for any of the fix up attributes you want RST to remove before sending.

In addition, you can change the Host ID, SOAP Host ID and SOAP EGM ID to test whether the host handles discrepencies in these values between the message and message wrapper.

About Fix Up Values

RST can modify, insert or delete (*fix up)* the three attributes in `g2s:g2sBody` as well as the class-level attributes of the G2S command. For each attribute:

- If a value is supplied by the user (see **Present** column below), RST uses that value.
- If a value is not supplied by the user, RST supplies the missing value.
- If the value is **-1**, the required attribute is removed from the message (missing).

For example:

Fix Up Rules

Attribute	Required	Present	Missing	-1
g2s:hostId	Yes	Use value	Supply argument value	Remove attribute
g2s:egmId	Yes	Use value	Supply current EGM	Remove attribute
g2s:dateTimeSent	Yes	Use value	Supply current date and time	Remove attribute
g2s:deviceId	Yes	Use value	Must be present	
g2s:dateTime	Yes	Use value	Supply current date and time	Remove attribute
g2s:commandId	Yes	Use value	Supply next command ID for the specified host in the method arguments.	Remove attribute
g2s:sessionType	No	Use value	Assume sessionType is G2S_request	Remove attribute
g2s:sessionId	No	Use value	If session type is G2S_request, add next session Id for the specified host in the message arguments.	Remove attribute
g2s:sessionRetry	No	Use value	Ignore	Remove attribute*

Attribute	Required	Present	Missing	-1
g2s:timeToLive	No	Use value	Ignore	Remove attribute*
g2s:sessionMore	No	Use value	Ignore	Remove attribute*
g2s:errorCode	No	Use value	Ignore	Remove attribute*
g2s:errorText	No	Use value	Ignore	Remove attribute*

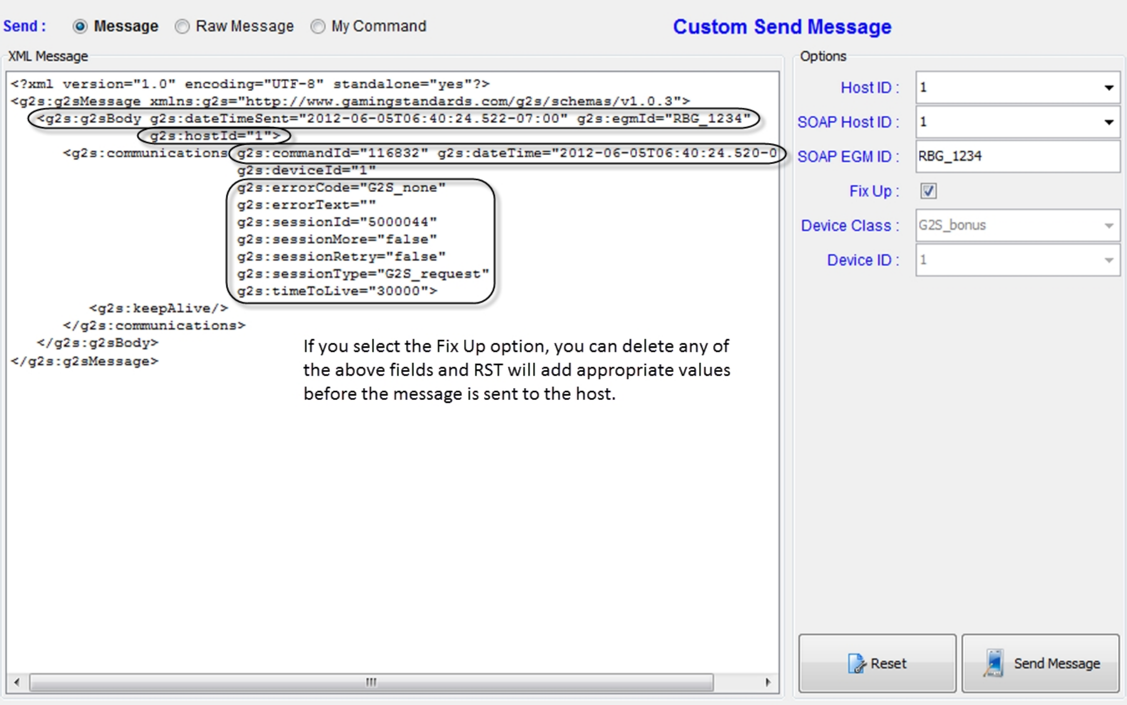
* This value is not needed because the missing attribute will be ignored.

The following is a Send Custom Message example sent from an EGM with the identifier **RBG_1234**:

- **Host ID** = 1
- **Fix Up** = True
- **EGM ID for SOAP** = [blank]
- **Host ID for SOAP** = [blank]
- **Message** =

To send a custom message:

1. From the Tester Toolkit layout, click **Custom Send Message**.
2. Verify that **Message** is selected in the **Send** field.
3. Type or paste the content you want to send in the XML Message text box. For this message type, you *must* include the entire message, including the device ID. If you select the Fix Up option, you can delete any or all of the fix up attributes, and RST will add appropriate values for the missing attributes\ before the message is sent to the host.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<g2s:g2sMessage xmlns:g2s="http://www.gamingstandards.com/g2s/schemas/v1.0.3">
   <g2s:g2sBody g2s:dateTimeSent="2012-06-05T06:40:24.522-07:00" g2s:egmId="RBG_1234"
                  g2s:hostId="1">
      <g2s:communications g2s:commandId="116832" g2s:dateTime="2012-06-05T06:40:24.520-0
                          g2s:deviceId="1"
                          g2s:errorCode="G2S_none"
                          g2s:errorText=""
                          g2s:sessionId="5000044"
                          g2s:sessionMore="false"
                          g2s:sessionRetry="false"
                          g2s:sessionType="G2S_request"
                          g2s:timeToLive="30000">
         <g2s:keepAlive/>
      </g2s:communications>
   </g2s:g2sBody>
</g2s:g2sMessage>
```

If you select the Fix Up option, you can delete any of the above fields and RST will add appropriate values before the message is sent to the host.

Host ID : 1
SOAP Host ID : 1
SOAP EGM ID : RBG_1234
Fix Up : ☑
Device Class : G2S_bonus
Device ID : 1

4. Modify the message **Options** as needed. Host ID, SOAP Host ID and SOAP EGM ID allow you to change the values to test how a host handles values that are inconsistent with the value(s) in the message.
   - **Host ID** - Click the drop-down arrow, and select the identifier of the recipient host.
   - **SOAP Host ID** - Click the drop-down arrow, and select the value for the SOAP host element.
   - **SOAP EGM ID** - Click the drop-down arrow, and select the value for the SOAP EGM ID element.
   - **Fix Up** - Select if you want RST to apply the fix up rules to the message you are sending.
   - **Device Class** - Indicates the recipient device class for the message. This field is not used for this type of custom message.
   - **Device ID** - Indicates the recipient device identifier for the message. This field is not used for this type of custom message.

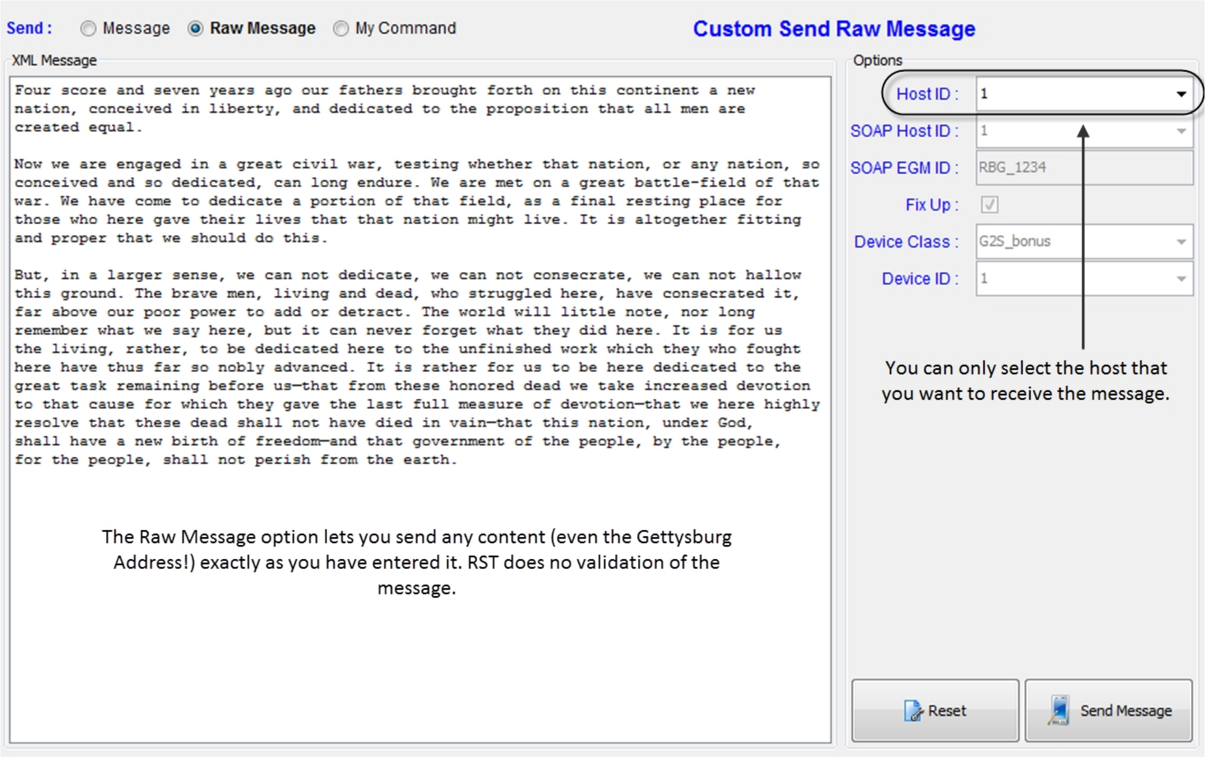
Note: You can reset the screen back to the default settings, including the removal of any XML Message text, by clicking **Reset**.

5. Click **Send Message** to send the message to the host.
6. Use the **Message Transcript** to verify that the message was sent and to view the host response.

Send a Raw Message

The Send Raw XML option lets you send any content to a host over a G2S connection. RST does no additional wrapping or validation of the message, but rather, sends exactly what you have entered to the host.

1. From the Tester Toolkit layout, click **Custom Send Message**.
2. Select **Raw Message** in the **Send** field.
3. Type or paste the content you want to send in the XML Message text box.



Custom Send Raw Message

Send : ○ Message ● Raw Message ○ My Command

XML Message

```
Four score and seven years ago our fathers brought forth on this continent a new
nation, conceived in liberty, and dedicated to the proposition that all men are
created equal.

Now we are engaged in a great civil war, testing whether that nation, or any nation, so
conceived and so dedicated, can long endure. We are met on a great battle-field of that
war. We have come to dedicate a portion of that field, as a final resting place for
those who here gave their lives that that nation might live. It is altogether fitting
and proper that we should do this.

But, in a larger sense, we can not dedicate, we can not consecrate, we can not hallow
this ground. The brave men, living and dead, who struggled here, have consecrated it,
far above our poor power to add or detract. The world will little note, nor long
remember what we say here, but it can never forget what they did here. It is for us
the living, rather, to be dedicated here to the unfinished work which they who fought
here have thus far so nobly advanced. It is rather for us to be here dedicated to the
great task remaining before us—that from these honored dead we take increased devotion
to that cause for which they gave the last full measure of devotion—that we here highly
resolve that these dead shall not have died in vain—that this nation, under God,
shall have a new birth of freedom—and that government of the people, by the people,
for the people, shall not perish from the earth.
```

The Raw Message option lets you send any content (even the Gettysburg Address!) exactly as you have entered it. RST does no validation of the message.

Options

Host ID : 1

SOAP Host ID : 1

SOAP EGM ID : RBG_1234

Fix Up : ☑

Device Class : G2S_bonus

Device ID : 1

You can only select the host that you want to receive the message.

4. Click the **Host ID** drop-down arrow under **Options**, and select the identifier of the recipient host. All other message configuration options are ignored for this message type.

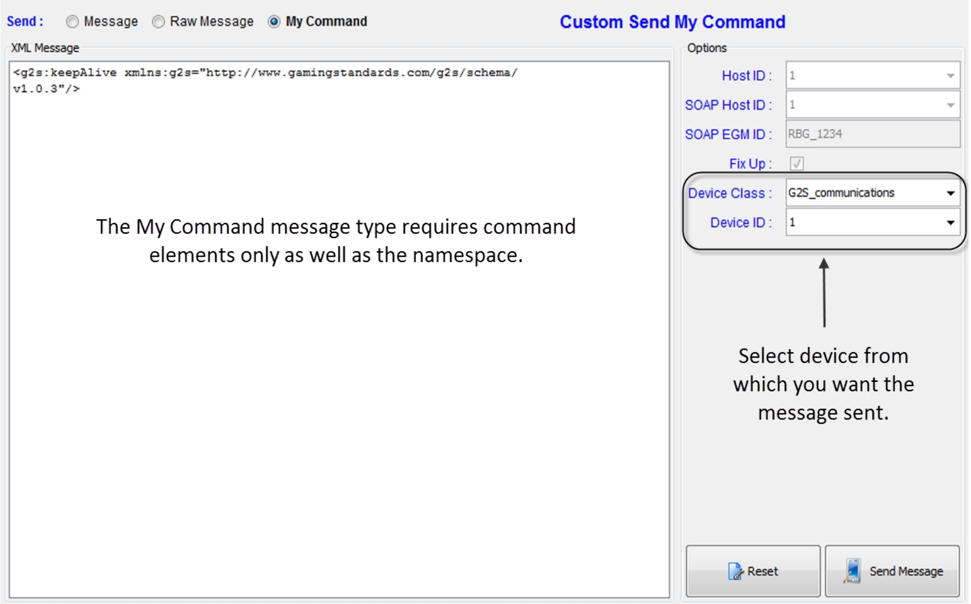
   Note: You can reset the screen back to the default settings, including the removal of any XML Message text, by clicking **Reset**.

5. Click **Send Message** to send the message to the host.
6. Use the **Message Transcript** to verify that the message was sent and to view the host response.

Send My Command

The Send My Command option lets you send a custom G2S command that RST then wraps with the G2S class and G2S message elements before sending. The G2S information completed automatically by RST includes command ID, session ID and date/time values. You can also add attributes and elements in custom name spaces, if required. This option is useful when you want to send a custom command from a device exposed through the descriptor list. At a minimum, you must include a command and associated namespace and ensure that the device class and ID includes the defined command.

1. From the Tester Toolkit layout, click **Custom Send Message**.
2. Select **My Command** in the **Send** field.
3. Type or paste the content you want to send in the XML Message text box.



Send : ○ Message ○ Raw Message ● My Command

Custom Send My Command

XML Message

```
<g2s:keepAlive xmlns:g2s="http://www.gamingstandards.com/g2s/schema/
v1.0.3"/>
```

Options

Host ID : 1

SOAP Host ID : 1

SOAP EGM ID : RBG_1234

Fix Up : ☑

Device Class : G2S_communications

Device ID : 1

The My Command message type requires command elements only as well as the namespace.

Select device from which you want the message sent.

4. Configure device to which you want the message sent. The message will be sent to the owner of the specified device.
   - **Device Class** - Click the drop-down arrow, and select the device class for the message.
   - **Device ID** - Click the drop-down arrow, and select the device identifier for the message.

   Note: You can reset the screen back to the default settings, including the removal of any XML Message text, by clicking **Reset**.
5. Click **Send Message** to send the message to the host.
6. Use the **Message Transcript** to verify that the message was sent and to view the host response.

Message Disruption

About Message Disruption

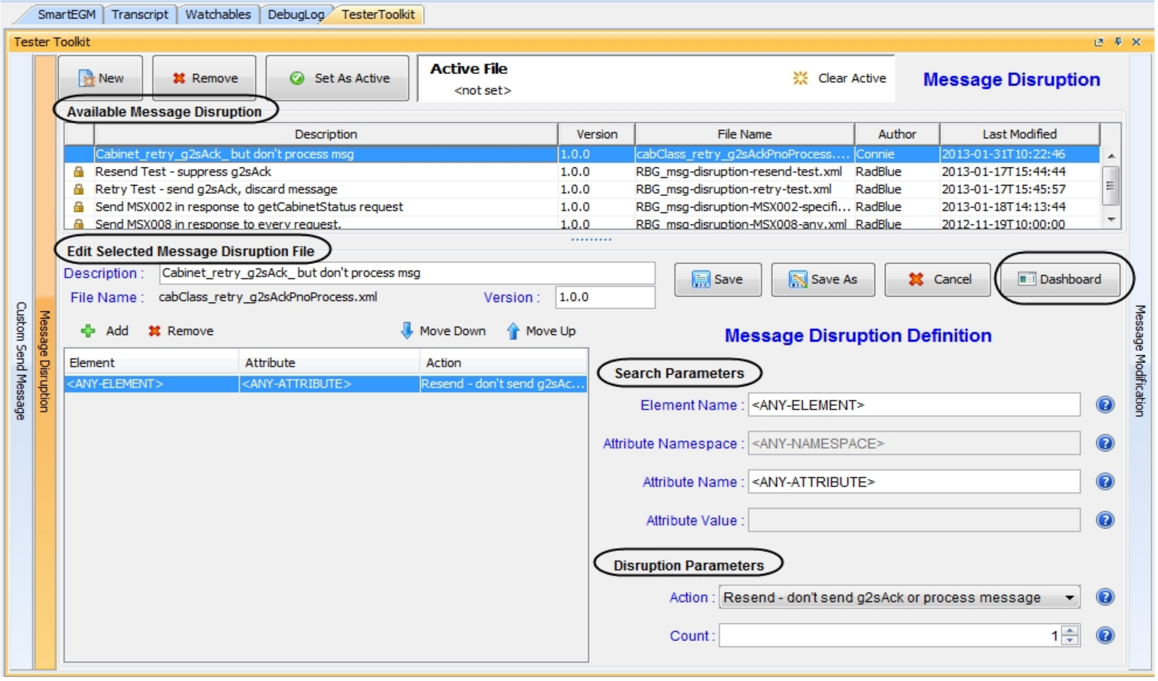
Message Disruption allows you to block or change the message response to the host using the RST predefined list of codes. You identify command elements and attributes to send to an EGM, and select a code to instruct the EGM how to respond.

These codes are listed in the **Disruption Parameters > Action** field, where you can select an action such as: retry, resend, undefined error, or choose one of the MSX001 - MSX008 error codes.

Using this feature, you create your new disruption message that becomes listed in the **Available Message Disruption** area, and then define:

- **Search parameters** - Those elements and attributes you want the RST to look for in each message.
- **Disruption parameters** - Those actions you want the RST to take and the number of times to take that action.

So, once defined, RST searches each message that it receives for the Element Name, if there is a match, RST searches for the Namespace (if one is defined), then the Attribute Name, and the Attribute Value. If all matches are found, RST repeats the action for the number of times (count) you specified.



	Description	Version	File Name	Author	Last Modified
	Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-01-31T10:22:46
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Description : Cabinet_retry_g2sAck_ but don't process msg

File Name : cabClass_retry_g2sAckPnoProcess.xml Version : 1.0.0

Element	Attribute	Action
<ANY-ELEMENT>	<ANY-ATTRIBUTE>	Resend - don't send g2sAc...

Element Name : <ANY-ELEMENT>

Attribute Namespace : <ANY-NAMESPACE>

Attribute Name : <ANY-ATTRIBUTE>

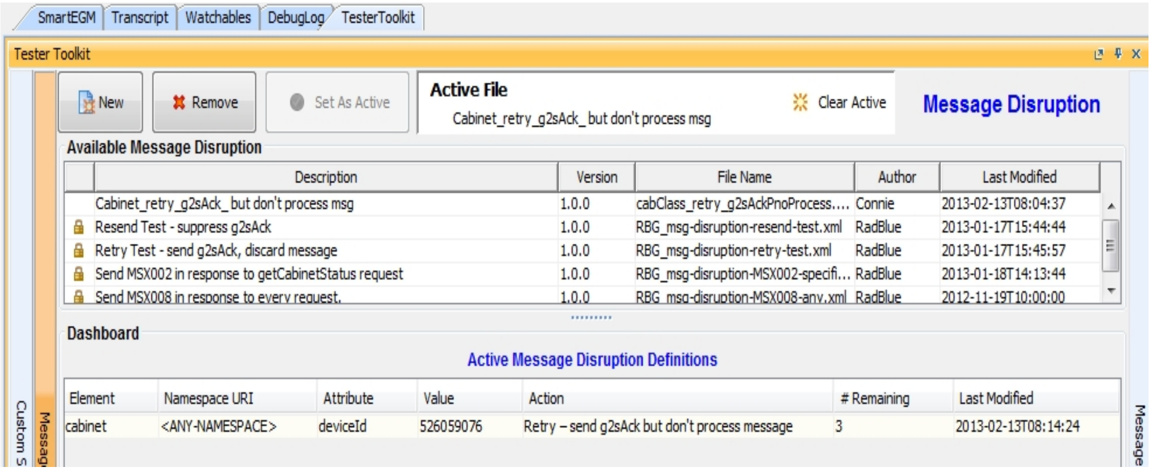
Attribute Value :

Action : Resend - don't send g2sAck or process message

Count : 1

The Message Disruption screen provides a dual purpose. Once you complete the Disruption definitions and activate your Disruption file, the **Edit Selected Message Disruption File** (bottom half of the screen) toggles to a Dashboard view.

The Dashboard, shows **Active Message Disruption Definitions** with remaining count. Each time the message is sent, the count decreases. The Dashboard is cleared when your active definition is complete.



Active File
Cabinet_retry_g2sAck_ but don't process msg

Available Message Disruption

	Description	Version	File Name	Author	Last Modified
	Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-02-13T08:04:37
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Dashboard

Active Message Disruption Definitions

Element	Namespace URI	Attribute	Value	Action	# Remaining	Last Modified
cabinet	<ANY-NAMESPACE>	deviceId	526059076	Retry – send g2sAck but don't process message	3	2013-02-13T08:14:24

How It Works

Follow this process to insert a disruptive message into your communications.

1. Create a new message disruption definition.
2. Define the message disruption definition by identifying the **Search** and **Disruption Parameters**.
3. Activate the new definition.

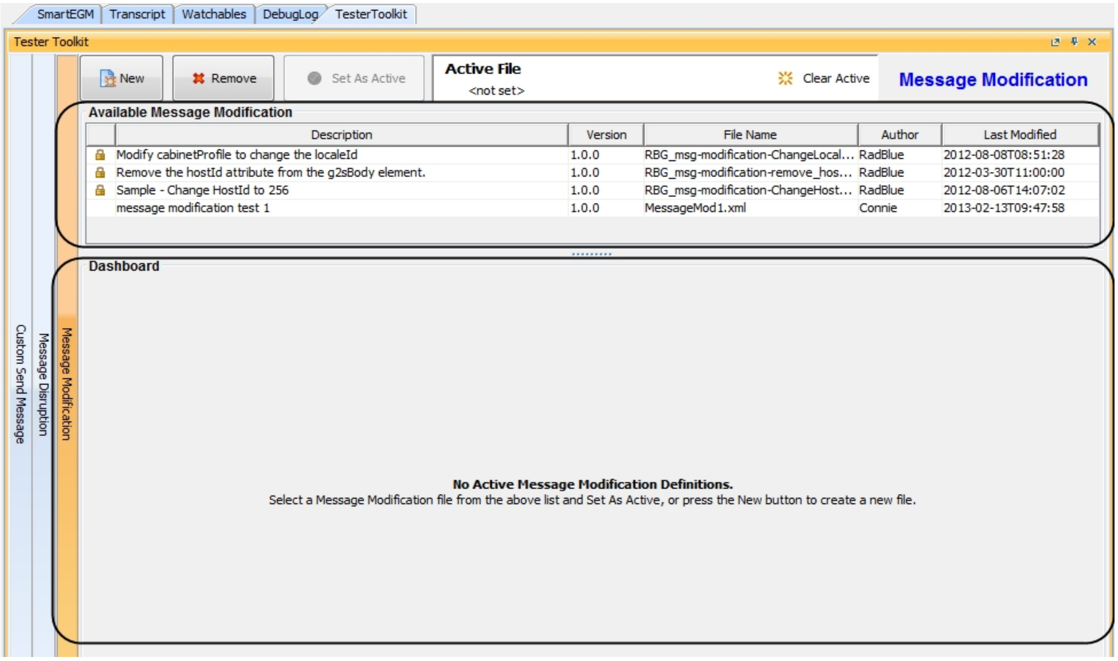
   Once you create, define, and activate a message disruption, the RST begins to match the search criteria against every message. RST begins with the Element Name. If there is a match, RST then matches the remaining attributes, in order.

   - If a match is found, RST starts the Action for the number of times you specified.
   - If a match is not found, it moves to the next message.
4. View the definition.

Create a New Message Disruption Definition

Follow these steps to create a new message disruption definition.

1. Go to **Tester Toolkit** tab and click the **Message Disruption** sidebar.

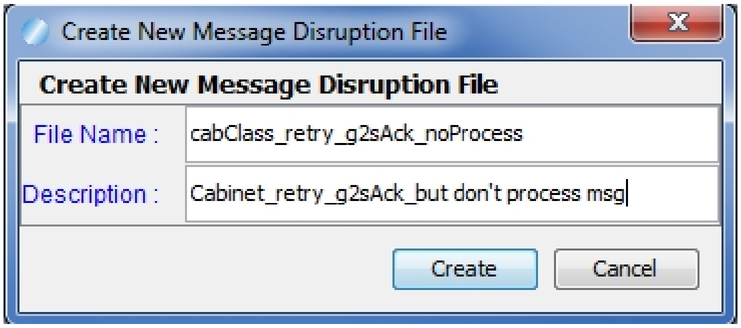


	Description	Version	File Name	Author	Last Modified
🔒	Modify cabinetProfile to change the localeId	1.0.0	RBG_msg-modification-ChangeLocal...	RadBlue	2012-08-08T08:51:28
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0	RBG_msg-modification-remove_hos...	RadBlue	2012-03-30T11:00:00
🔒	Sample - Change HostId to 256	1.0.0	RBG_msg-modification-ChangeHost...	RadBlue	2012-08-06T14:07:02
	message modification test 1	1.0.0	MessageMod1.xml	Connie	2013-02-13T09:47:58

No Active Message Modification Definitions.
Select a Message Modification file from the above list and Set As Active, or press the New button to create a new file.

2. Click **New** to create a new message disruption definition.

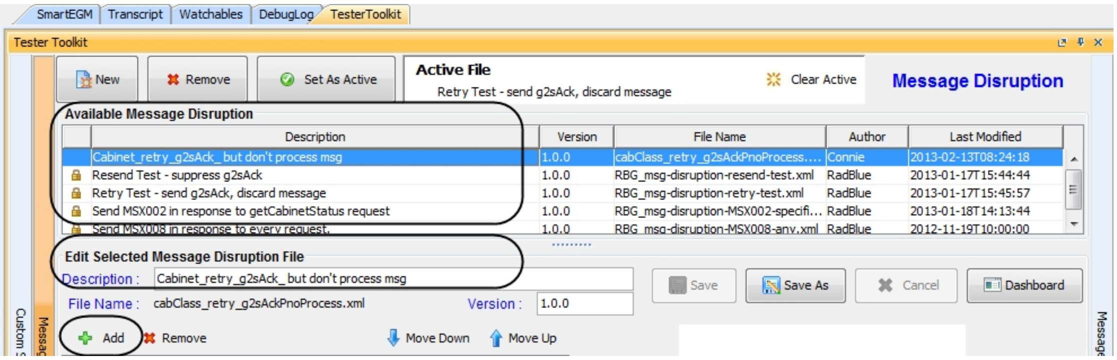
3. At the pop-up type the **file name** (no spaces) and **description** of the test.


Create New Message Disruption File
X
Create New Message Disruption File
Create
Cancel

File Name : cabClass_retry_g2sAck_noProcess

Description : Cabinet_retry_g2sAck_but don't process msg

4. Click **Create**.



Active File
Retry Test - send g2sAck, discard message

Available Message Disruption

	Description	Version	File Name	Author	Last Modified
	Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-02-13T08:24:18
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Edit Selected Message Disruption File

Description : Cabinet_retry_g2sAck_ but don't process msg

File Name : cabClass_retry_g2sAckPnoProcess.xml Version : 1.0.0

5. You see your new file automatically listed in two areas of the screen:

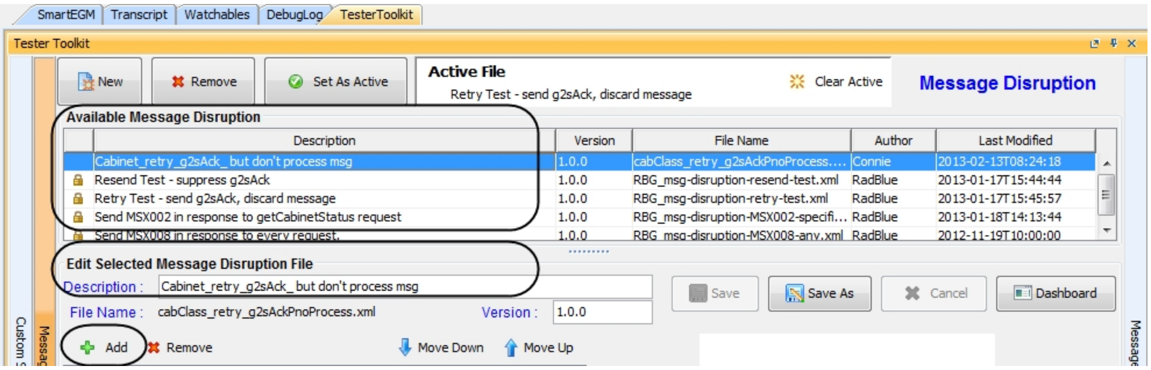
- **Available Message Disruption** - A lock in front of a file indicates that it is read-only and cannot be deleted. If you choose a locked file as a template and make changes, you must use **Save As**.
- **Edit Selected Message Disruption File/Dashboard** - This panel toggles from dashboard to edit screens.

Note: You can change existing Disruption fields — fix misspellings or cryptic titles, or change Search/Disruption parameters. See Edit a Definition.

Define the Message Disruption Definition

Follow these steps to define the search parameters.

1. In the **Available Message Disruption** panel, highlight a description. Here we selected **Cabinet_retry_g2sAck_but don't process msg**.



Active File
Retry Test - send g2sAck, discard message

Available Message Disruption

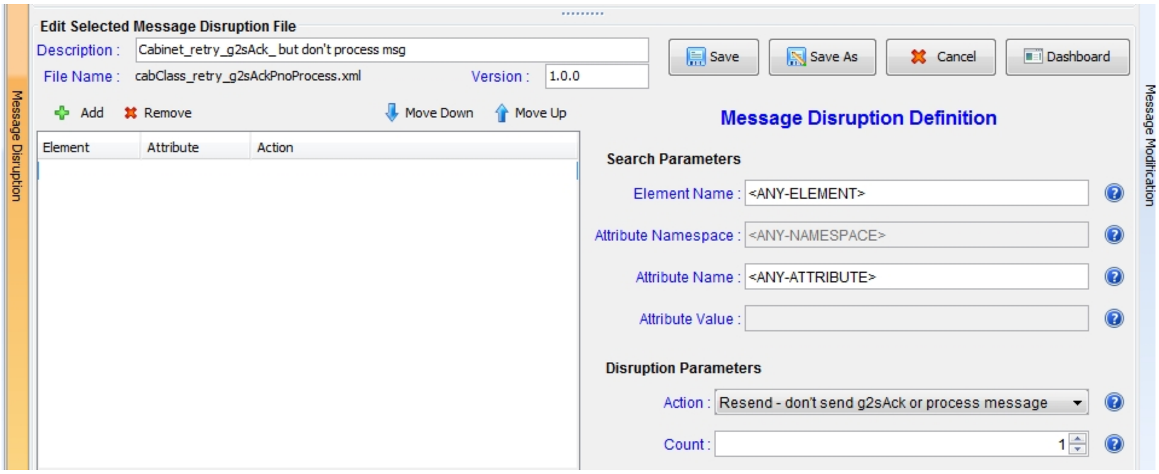
Description	Version	File Name	Author	Last Modified
Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-02-13T08:24:18
Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Edit Selected Message Disruption File

Description : Cabinet_retry_g2sAck_ but don't process msg

File Name : cabClass_retry_g2sAckPnoProcess.xml Version : 1.0.0

2. Click **Add** to open an existing, or the new message disruption, to see the Definitions fields.



Edit Selected Message Disruption File

Description : Cabinet_retry_g2sAck_ but don't process msg

File Name : cabClass_retry_g2sAckPnoProcess.xml Version : 1.0.0

Message Disruption Definition

Search Parameters

Element Name : <ANY-ELEMENT>

Attribute Namespace : <ANY-NAMESPACE>

Attribute Name : <ANY-ATTRIBUTE>

Attribute Value :

Disruption Parameters

Action : Resend - don't send g2sAck or process message

Count : 1

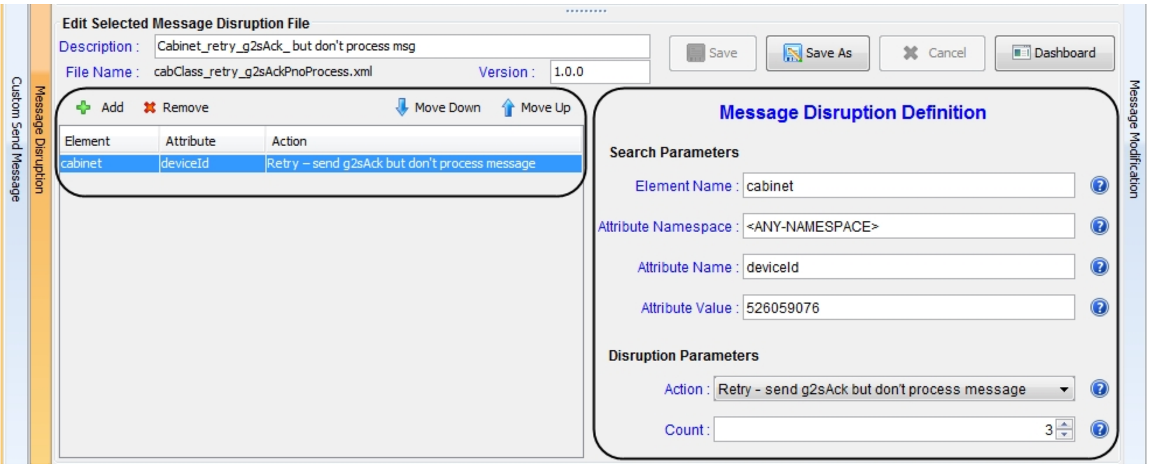
3. Use the **Message Disruption Definition** fields to define the disruption. Do not use quotes or namespace prefixes in any field.

   - **Element Name -** Type the element name to search for, typically a class or a command, regardless of namespace. In our example, we selected **cabinet.**

     The element name is always required, either a real value or a wildcard.
     Wildcard selections are:

     - <ANY-ELEMENT> - Matches any class element, such as `noteAcceptor`.
     - <ANY-COMMAND> - Matches any command element, such as `g2sBody`.

   - **Attribute Namespace -** Type the namespace of the attribute.
     The wildcard <ANY-NAMESPACE> searches for any namespace. We selected the wildcard.



Edit Selected Message Disruption File

Description : Cabinet_retry_g2sAck_ but don't process msg

File Name : cabClass_retry_g2sAckPnoProcess.xml Version : 1.0.0

Element	Attribute	Action
cabinet	deviceId	Retry – send g2sAck but don't process message

Message Disruption Definition

Search Parameters

Element Name : cabinet

Attribute Namespace : <ANY-NAMESPACE>

Attribute Name : deviceId

Attribute Value : 526059076

Disruption Parameters

Action : Retry - send g2sAck but don't process message

Count : 3

- **Attribute Name** - Type the name of the attribute. In our example, **deviceId**.

Note: If you use the default <ANY ATTRIBUTE> (the wildcard) then both Namespace and Value fields are disabled because that configuration does not present a valid search.

- **Attribute Value** - Type the value of the attribute to match (* = wildcard). In our example, we selected **526059076**.
- **Action** - Choose the action you want to happen when the attributes match.

  In our example, we selected **Retry - send g2sAck but don't process message**.
- **Count** - Type or select the number of times the specified definition is applied. We typed **3**.

2. Click **Save**. You now see your message listed in the Element, Attribute, Action box (under the Add button).

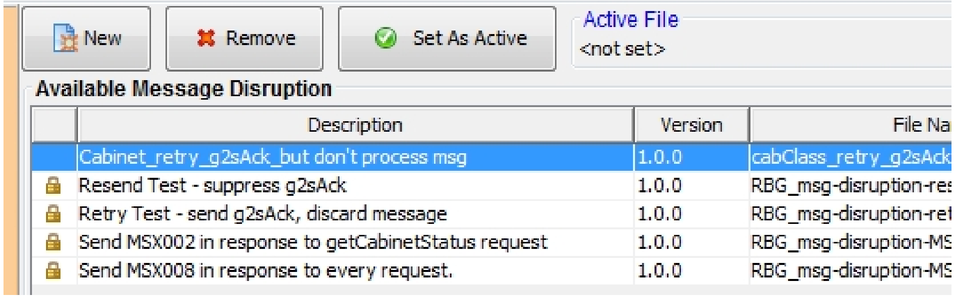
   Note: Once Set as Active, RST searches each message that it receives for the Element Name. If there is a match, it searches for the Namespace (if one is defined). Then, RST searches for the Attribute Name and Attribute Value. If all matches are found, RST starts the Action for the number of times you specify.

   In our screen example, when **deviceId** matches the value (**526059076**), RST sends the error message **Retry-send g2sAck but don't process message**.
3. You can now choose to activate this disruption file or create another definition.

Activate the Message Disruption Definition

Follow these steps to activate your Message Disruption.

1. In the **Available Message Disruption** panel > **Description**, click to highlight your message.
2. Click **Set As Active**.
3. At the pop-up box, click **OK**.

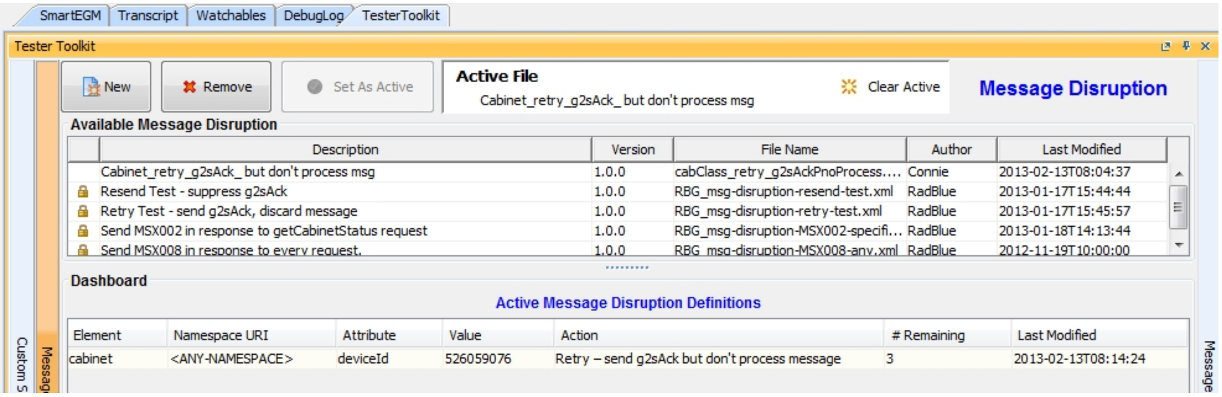


Active File
<not set>

Available Message Disruption

	Description	Version	File Na
	Cabinet_retry_g2sAck_but don't process msg	1.0.0	cabClass_retry_g2sAck
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-res
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-ret
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MS
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MS

4. Confirm that the **Active File** changed from **<not set>** to the name of your file.



Active File
Cabinet_retry_g2sAck_ but don't process msg

Available Message Disruption

	Description	Version	File Name	Author	Last Modified
	Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-02-13T08:04:37
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Dashboard

Active Message Disruption Definitions

Element	Namespace URI	Attribute	Value	Action	# Remaining	Last Modified
cabinet	<ANY-NAMESPACE>	deviceId	526059076	Retry – send g2sAck but don't process message	3	2013-02-13T08:14:24

5. Notice the Dashboard appears showing the active message being sent for the number of times you selected.

   Note: Click the **Clear Active** button to remove the Active file.

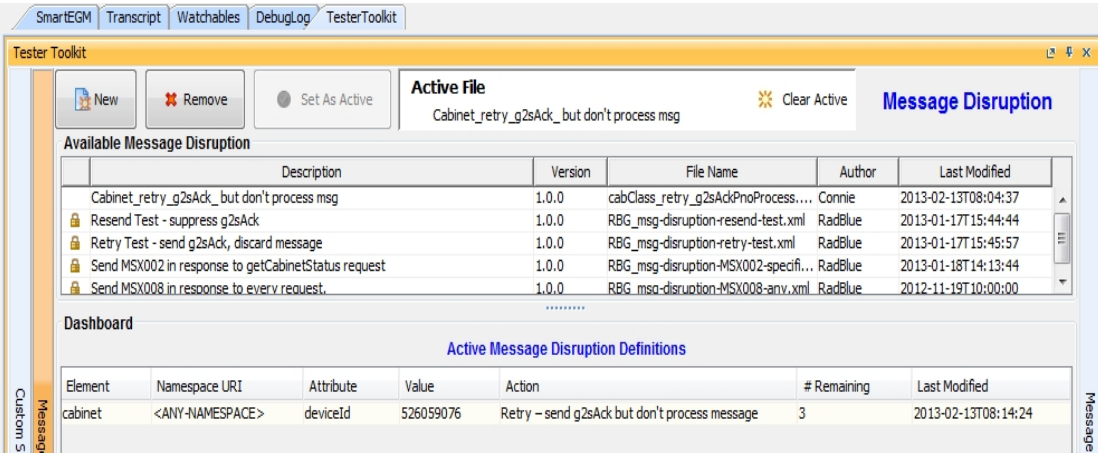
View the Active Disruption File

The Dashboard, shows **Active Message Disruption Definitions** with remaining count.

- If your **Count** entry was two or higher, then in the **# Remaining** column, you would see the number flashing for three seconds, and the number decreasing as the action takes place.
- The active definition files are removed automatically from the list when the remaining count gets to zero.

- The **Active Message Dashboard** is cleared, when all active definitions are completed.

You can set the Active File again, or edit the definition.



Active File
Cabinet_retry_g2sAck_ but don't process msg

Available Message Disruption

	Description	Version	File Name	Author	Last Modified
	Cabinet_retry_g2sAck_ but don't process msg	1.0.0	cabClass_retry_g2sAckPnoProcess....	Connie	2013-02-13T08:04:37
🔒	Resend Test - suppress g2sAck	1.0.0	RBG_msg-disruption-resend-test.xml	RadBlue	2013-01-17T15:44:44
🔒	Retry Test - send g2sAck, discard message	1.0.0	RBG_msg-disruption-retry-test.xml	RadBlue	2013-01-17T15:45:57
🔒	Send MSX002 in response to getCabinetStatus request	1.0.0	RBG_msg-disruption-MSX002-specifi...	RadBlue	2013-01-18T14:13:44
🔒	Send MSX008 in response to every request.	1.0.0	RBG_msg-disruption-MSX008-any.xml	RadBlue	2012-11-19T10:00:00

Dashboard

Active Message Disruption Definitions

Element	Namespace URI	Attribute	Value	Action	# Remaining	Last Modified
cabinet	<ANY-NAMESPACE>	deviceId	526059076	Retry – send g2sAck but don't process message	3	2013-02-13T08:14:24

Edit a Definition

Follow these steps to edit a definition.

1. In the **Available Message Disruption** panel, click a definition to edit.
2. At this point, you can do any of the following:
   - Change the Search or Disruption parameters as needed.
   - Fix a misspelled or cryptic title (in the Description field).
   - Change the Version (in the Version field).
3. Once you begin typing or using the backspace, the **Save**/**Save As** buttons activate.
4. Click **Save**.
5. In the **Available Message Disruption** panel, click the newly modified definition.
6. Click **Set as Active** to activate the changed definition.
7. Click **Dashboard** to see the active message and the number of times the message is sent.

Remove a Definition

Follow these steps to remove a definition.

1. Go to **Available Message Disruption** panel > **Description**.
2. Click to highlight the definition you want to remove.
3. Click **Remove**.
4. At the pop-up, select **Yes**.

Modify Outgoing Messages

About Message Modification

Message modification lets you define a list of changes that RST will make to outgoing G2S messages. Message changes can include changing an attribute, deleting an attribute or adding a new attribute.

RST maintains a first-in, first-out list of modifications. For each G2S message RST sends, RST checks whether the message matches one or more modifications in the list. As the RST matches modifications, it decrements the modification's count. Note that if the count is **-1**, RST will always perform the specified modification. When the modification's count reaches zero, that modification is removed from list of modifications.

When RST processes a message, it first iterates over the entire list of modifications, looking for all modifications that modify the `g2s:g2sBody` element. Since there can more than one G2S command in one message, RST checks all of the class-level elements (child elements of the `g2s:g2sBody` element) next. For each class-level element, RST checks the entire list of modifications looking for matches. When a modification matches, RST either sets (SET) an attribute value (adds if not present) or removes (REMOVE) an attribute. For example:

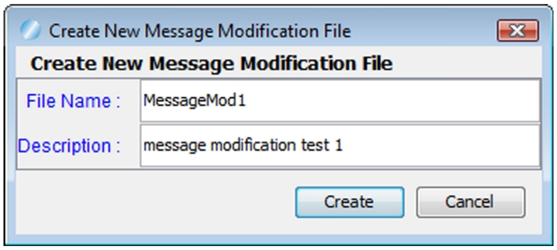
Initial List of Modifications

ID	Action	Element	Attribute Namespace	Attribute Name	Attribute Value	Count
1	SET	wat	http://www.radblue.com/	special		-1
2	REMOVE	g2sBody	http://www.gamingstandards.com/g2s/schemas/v1.0.3	dateTimeSent		1
3	SET	g2sBody	http://www.gamingstandards.com/g2s/schemas/v1.0.3	egmId	Name	2
4	SET	eventReport	http://www.gamingstandards.com/g2s/schemas/v1.0.3	eventId	ALPHA	1
5	REMOVE	commsOnLine	http://www.gamingstandards.com/g2s/schemas/v1.0.3	egmLocation		1

Create a New Message Modification File

From the Message Modification screen on the Tester Toolkit layout, you can create as many message modifications as you need. Each message modification file contains changes to message attributes for a defined class or command. When a message modification file is set to active, RST automatically makes the changes specified in the file to each applicable outgoing message.

1. From the Tester Toolkit layout, click **Message Modification**.
2. Click **New**.

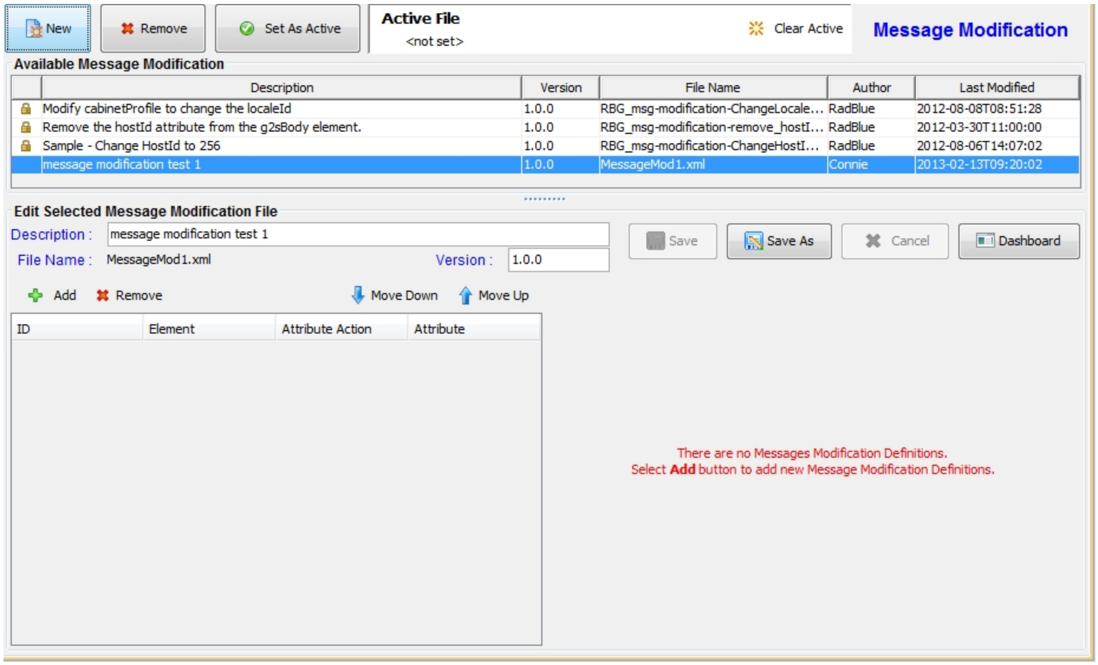


Create New Message Modification File
Create New Message Modification File
Create
Cancel

File Name : MessageMod1

Description : message modification test 1

3. Type a file name and file description for the new message modification file.
4. Click **Create**.



Active File <not set>

	Description	Version	File Name	Author	Last Modified
🔒	Modify cabinetProfile to change the localeId	1.0.0	RBG_msg-modification-ChangeLocale…	RadBlue	2012-08-08T08:51:28
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0	RBG_msg-modification-remove_hostI…	RadBlue	2012-03-30T11:00:00
🔒	Sample - Change HostId to 256	1.0.0	RBG_msg-modification-ChangeHostI…	RadBlue	2012-08-06T14:07:02
	message modification test 1	1.0.0	MessageMod1.xml	Connie	2013-02-13T09:20:02

Description : message modification test 1

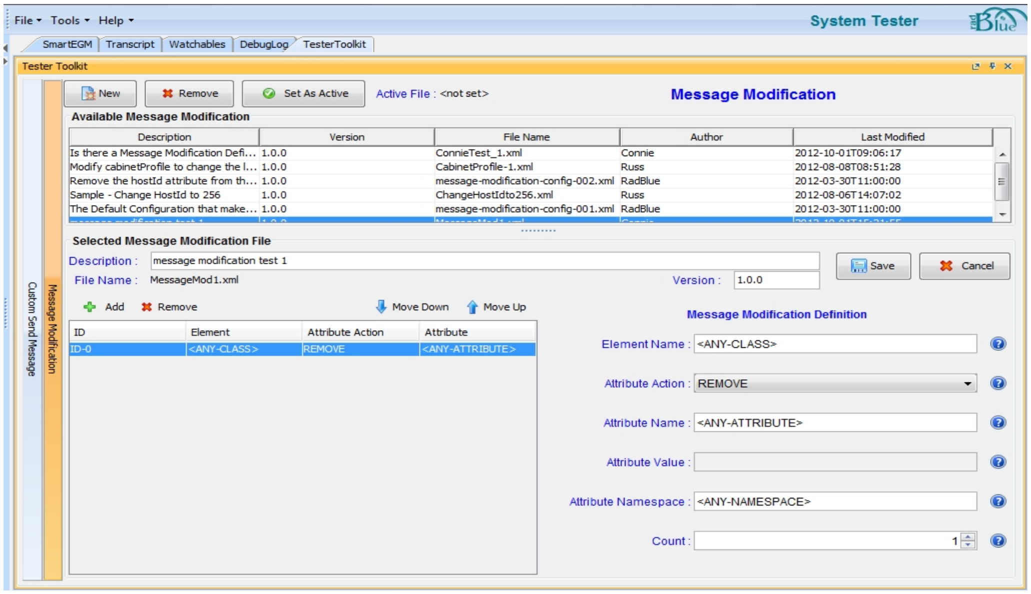
File Name : MessageMod1.xml

Version : 1.0.0

ID	Element	Attribute Action	Attribute

There are no Messages Modification Definitions.
Select **Add** button to add new Message Modification Definitions.

5. Click **Add** to add an attribute to the message modification file.



Active File : <not set>

Message Modification

Available Message Modification

Description	Version	File Name	Author	Last Modified
Is there a Message Modification Defi…	1.0.0	ConnieTest_1.xml	Connie	2012-10-01T09:06:17
Modify cabinetProfile to change the l…	1.0.0	CabinetProfile-1.xml	Russ	2012-08-08T08:51:28
Remove the hostId attribute from th…	1.0.0	message-modification-config-002.xml	RadBlue	2012-03-30T11:00:00
Sample - Change HostId to 256	1.0.0	ChangeHostIdto256.xml	Russ	2012-08-06T14:07:02
The Default Configuration that make…	1.0.0	message-modification-config-001.xml	RadBlue	2012-03-30T11:00:00

Selected Message Modification File

Description : message modification test 1

File Name : MessageMod1.xml

Version : 1.0.0

ID	Element	Attribute Action	Attribute
ID-0	<ANY-CLASS>	REMOVE	<ANY-ATTRIBUTE>

Message Modification Definition

Element Name : <ANY-CLASS>

Attribute Action : REMOVE

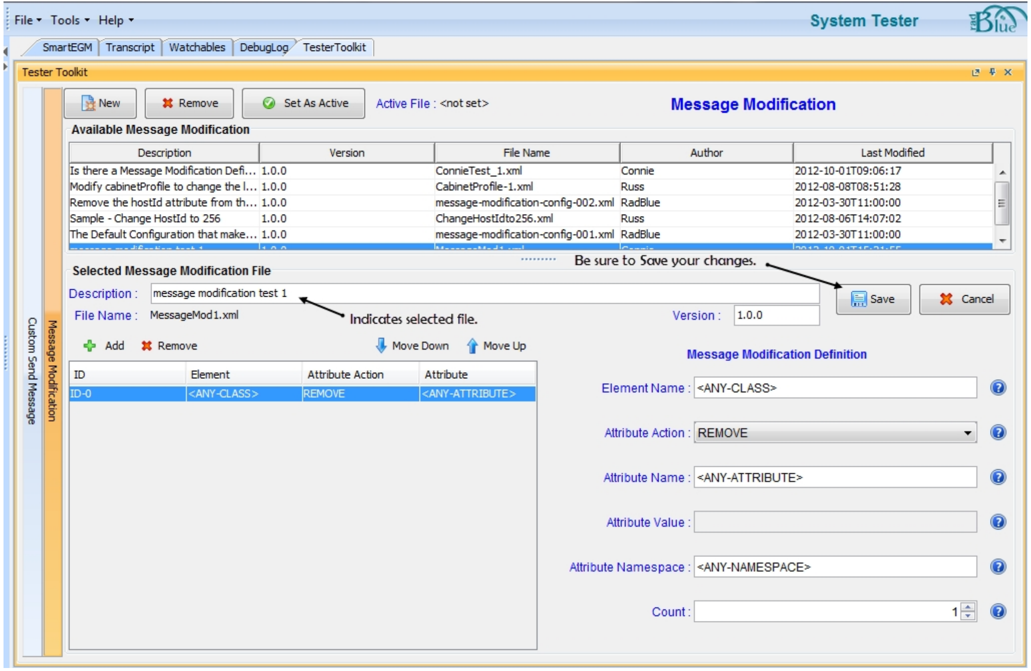
Attribute Name : <ANY-ATTRIBUTE>

Attribute Value :

Attribute Namespace : <ANY-NAMESPACE>

Count : 1

6. Use the **Message Modification Definition** fields to customize the message.



Active File : <not set>

Message Modification

Available Message Modification

Description	Version	File Name	Author	Last Modified
Is there a Message Modification Defi…	1.0.0	ConnieTest_1.xml	Connie	2012-10-01T09:06:17
Modify cabinetProfile to change the l…	1.0.0	CabinetProfile-1.xml	Russ	2012-08-08T08:51:28
Remove the hostId attribute from th…	1.0.0	message-modification-config-002.xml	RadBlue	2012-03-30T11:00:00
Sample - Change HostId to 256	1.0.0	ChangeHostIdto256.xml	Russ	2012-08-06T14:07:02
The Default Configuration that make…	1.0.0	message-modification-config-001.xml	RadBlue	2012-03-30T11:00:00

Be sure to Save your changes.

Selected Message Modification File

Description : message modification test 1

Indicates selected file.

File Name : MessageMod1.xml

Version : 1.0.0

ID	Element	Attribute Action	Attribute
ID-0	<ANY-CLASS>	REMOVE	<ANY-ATTRIBUTE>

Message Modification Definition

Element Name : <ANY-CLASS>

Attribute Action : REMOVE

Attribute Name : <ANY-ATTRIBUTE>

Attribute Value :

Attribute Namespace : <ANY-NAMESPACE>

Count : 1

- **Element Name -** Type the name of the element to search for, typically `g2sBody`, a class, or a command (for example, **noteAcceptor**).

  The special value **<ANY-CLASS>** matches any class level element, regardless of namespace.

  The special value **<ANY-COMMAND>** matches any command level element, regardless of namespace.

Note: RST search for the first matching element in each message and, if a match is found, acts on that element. If an attribute is acted on within that element, the search ends. If no action is taken, RST continues to search for an instance of the element until the end of the message is reached or an action is taken.

- **Attribute Action** - Click the drop-down arrow, and choose to **set** or **remove** the specified attribute.
- **Attribute Name** - Type the name of the attribute for adding and searching (for example, **noteValueInEscrow**).
- **Attribute Value** - If the `action` element is **set**, this is the new value of the attribute, whether it is added or replaced. If the `action` element is **remove**, the value of this element is ignored.
- **Attribute Namespace** - Namespace of the attribute, for both adding and searching. When you use `<ANY-NAMESPACE>`, all attributes are searched regardless of namespace. In a set command, the namespace can be specified, or else it will be set to the namespace of the matching element.
- **Count** - Type or select the number of times the specified definition is applied. When the count reaches zero, this modification record is deleted by RST.

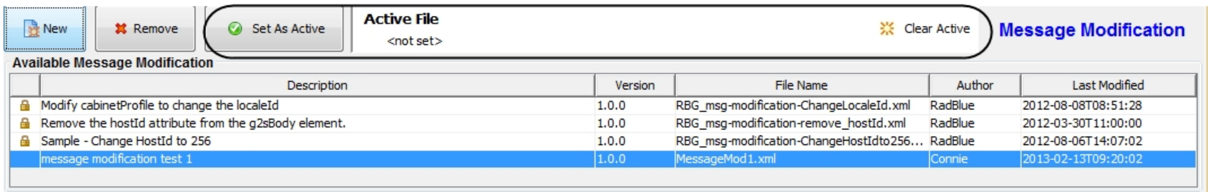
6. Click **Save** to retain the changes to the selected message modification file.

   Note that quotes are not needed in any field, and namespace prefixes are not used in any field.

Set or Clear the Active Message Modification File

From the Message Modification screen on the Tester Toolkit layout, you can set and clear the active message modification file. Once set, the active message modification file changes are applied to all applicable outgoing messages.

1. From the Tester Toolkit layout, click **Message Modification**.
2. Click to select the message modification file you want to make active.

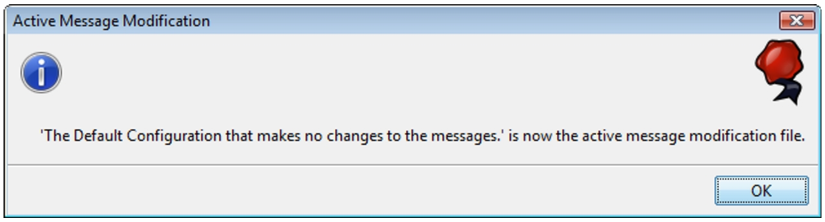


Active File
<not set>

Available Message Modification

	Description	Version	File Name	Author	Last Modified
🔒	Modify cabinetProfile to change the localeId	1.0.0	RBG_msg-modification-ChangeLocaleId.xml	RadBlue	2012-08-08T08:51:28
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0	RBG_msg-modification-remove_hostId.xml	RadBlue	2012-03-30T11:00:00
🔒	Sample - Change HostId to 256	1.0.0	RBG_msg-modification-ChangeHostIdto256…	RadBlue	2012-08-06T14:07:02
	message modification test 1	1.0.0	MessageMod1.xml	Connie	2013-02-13T09:20:02

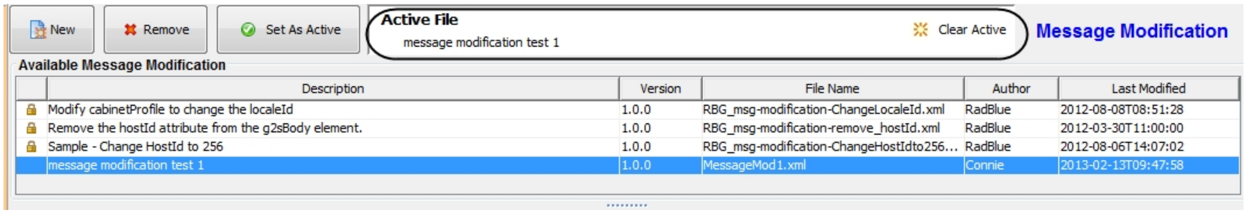
3. Click **Set As Active**.



Active Message Modification

'The Default Configuration that makes no changes to the messages.' is now the active message modification file.

4. Click **OK**.



Active File
message modification test 1

Available Message Modification

	Description	Version	File Name	Author	Last Modified
🔒	Modify cabinetProfile to change the localeId	1.0.0	RBG_msg-modification-ChangeLocaleId.xml	RadBlue	2012-08-08T08:51:28
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0	RBG_msg-modification-remove_hostId.xml	RadBlue	2012-03-30T11:00:00
🔒	Sample - Change HostId to 256	1.0.0	RBG_msg-modification-ChangeHostIdto256…	RadBlue	2012-08-06T14:07:02
	message modification test 1	1.0.0	MessageMod1.xml	Connie	2013-02-13T09:47:58

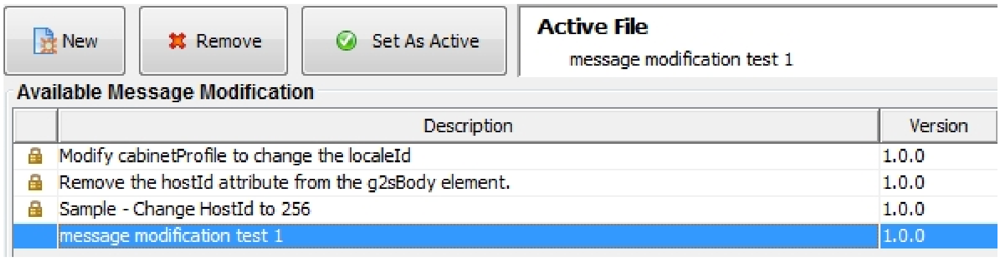
RST will apply the new message modification file attributes immediately.

5. To set a different message modification file as active, repeat steps 2-3.
6. To clear a message modification file so RST has no active file, click **Clear Active**.

Delete a Message Modification File

From the Message Modification screen on the Tester Toolkit layout, you can delete a message modification file as needed.

1. From the Tester Toolkit layout, click **Message Modification**.
2. Click to select the message modification file in the **Available Message Modification** list that you want to delete.

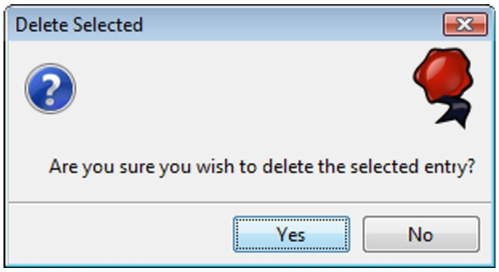


Active File
message modification test 1

Available Message Modification

	Description	Version
🔒	Modify cabinetProfile to change the localeId	1.0.0
🔒	Remove the hostId attribute from the g2sBody element.	1.0.0
🔒	Sample - Change HostId to 256	1.0.0
	message modification test 1	1.0.0

3. Click **Remove**.


Delete Selected

Are you sure you wish to delete the selected entry?

4. Click **Yes** to delete the file from RST.

About the SmartEGM Configuration File

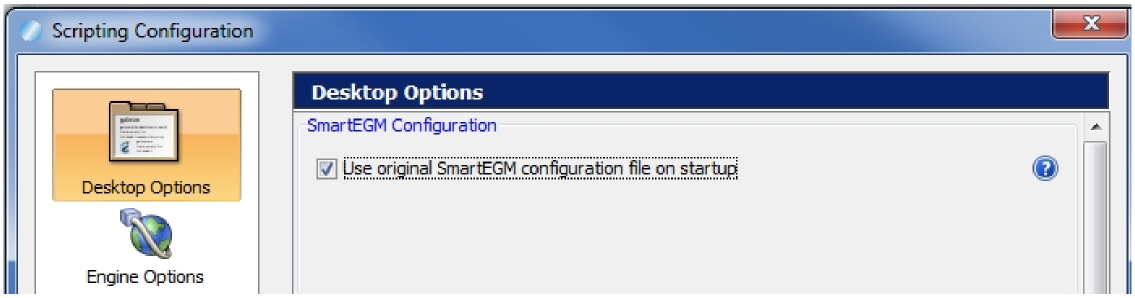
The SmartEGM configuration file is the data model for RST. To use RST, you must load a valid SmartEGM configuration file. You can use one of the default SmartEGM configuration files or customize a SmartEGM file to suit your testing needs. The SmartEGM file contains:

- RST data model information (supported devices, meters, events and options)
- G2S schema that will be used during testing
- Host information (identifiers, URL, description)

You can select to use a default original**SmartEGM configuration file** on startup.

From the menu bar, go to **Tools > Configure > Desktop Options.**

- If **selected**,
  - RST automatically loads a nonupdated version of the SmartEGM configuration file each time you restart the tool.
  - This file contains default files from which you can select a configuration. See the list of default files.
- If this checkbox is **cleared**, the new values are used when RST is restarted.



☑ Use original SmartEGM configuration file on startup

Configuration Default Files

The RST provides a SmartEGM configuration default file that you can select each time you start the RST. Go to **Tools >Configure > Desktop Options**.

Use original SmartEGM configuration file on startup

- If selected, RST automatically loads a nonupdated version of the SmartEGM configuration file on startup.
- If not selected, the RST uses the new values you entered when restarted.

Change Configuration—Default Files

The **Change SmartEGM Configuration** button (under **SmartEGM Control**) takes you to the smart-egm directory in the installation folder, which contains all default SmartEGM configuration files. This table lists the default configurations available for different tests you may wish to run. Each SmartEGM configuration file contains all available devices and attributes.

To view or modify the XML configuration file for each of the default files, load the file and go to [**Install Directory**] **> RST > smart-conf > smart-egm.**

If you change configuration files, you must **Restart** the RST.

xml File smartegm-config-gsa	Description
-gsa	The master configuration file. All other configuration files are variations of one. Used to communicate with RGS.
-am	Enables Audit Meters.
-central	Enables Play Central Game. This file includes two central game play devices that correspond to game play devices on the EGM.
-rpa	Enables communication with RPA.
-updated	Stores the changes you make to the EGM ID, Host ID, and/or Host URL.
-no-namespace	Disables namespace negotiation (per G2S 1.1.). If you want to disable namespace negotiation in another SmartEGM configuration file (for example, smartegm-config-ip), go to the edm:communications element and change the edm:suppress-namespace-negotiation attribute to "true"so that the negotiate namespaces attribute is not set in `commsOnLine` command.
-two-hosts	Allows two hosts to connect to RST at once. By default, the defined hosts are RGS 1 and RGS 2. However, you can change one or both hosts as required for testing.
-egm-2	RST installs, two separate instances. There is a separate set of SmartEGM configuration files for each RST instance. Files for the second RST instance are appended with -2.
-ip	Enables informed player extensions.

xml File smartegm-config-gsa	Description
-student-edition	Use with student licenses only to communicate with RGS.
-student-edition-rpa	Use with student licenses only to communicate with RPA.

Edit 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.
   - editing the host identifier list, see Adding or Modifying Hosts.
   - adding game play devices, see Adding GamePlay Devices.
   - modifying progressives, see Modifying Progressive Devices.
   - adding unsupported events, see Adding Unsupported Events Elements.
   - configuring offline ID validation, see Adding Offline Validation IDs.
   - resetting the command ID sequence, see Resetting Command ID Sequence.
   - adding game outcome, see Configure Game Outcome.
   - configuring Audit Meters, see Configure Audit Meters.
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.

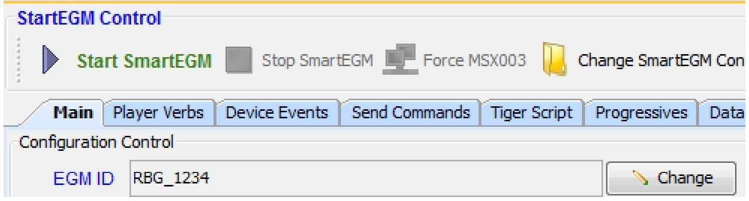
Edit the EGM ID

You can modify the SmartEGM EGM identifier in two ways:

- Through the RST interface
- Through the XML file

RST Interface

1. At the Main tab, next to the EGM ID field, click the **Change** button.



EGM ID RBG_1234

2. At the Change SmartEGM ID box, type a new **EGM ID** following pattern [A-Z0-9](3)_.*
3. Click **OK**.
4. Look at the new identifier in the EGM ID field.

XML File

1. Go to **InstallDirectory\smart-conf\smart-egm**
2. 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">
```

3. **Save** and **close** the file before loading it into RST.

Add or Modify Hosts

You can modify the SmartEGM's registered host list in two ways

- Using the RST Interface
- Using the .xml file

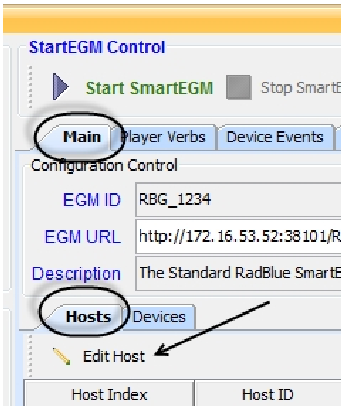
The following table shows the default host list.

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

Edit a Host ID and URL

Follow these steps to edit the host system identifier, description and URL defined in the currently loaded SmartEGM configuration file.

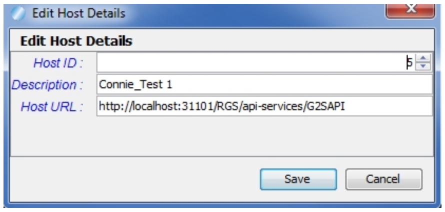
1. From the Main tab,
   - click the **Hosts** tab
   - click the **Edit Host** button.



EGM ID RBG_1234
EGM URL http://172.16.53.52:38101/R
Description The Standard RadBlue SmartE

2. Type or select the new Host ID. Here we typed 5.
3. Type a new host description..
4. Type a new host URL.

5. Click **Save** — The SmartEGM is automatically stopped, and the updated configuration file is loaded.



Host ID : 5

Description : Connie_Test 1

Host URL : http://localhost:31101/RGS/api-services/G2SAPI

7. Look for your new configuration listed in the columns below the Edit Host button.
8. Click **Start SmartEGM** to restart the SmartEGM.

   Note: When you Exit RST—ending your testing session—your new configuration is listed in the updated.xml configuration file.

Edit a Host Using the XML File

1. Go to: **Install Directory\smart-conf\smart-egm**
2. Type your changes in any of the attributes:
   - **edm:host-id**
   - **edm:url**
   - **edm:description**
   - **edm:host-registered**

   Each registered host ID must be unique and contain a valid URL (unless it is an empty row, signified by **host-id =”0”**).

   Note: The *hostIndex* required attribute is handled automatically by RST, so you do not need to enter it when adding new host entries to the file.

```
<?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" />
```

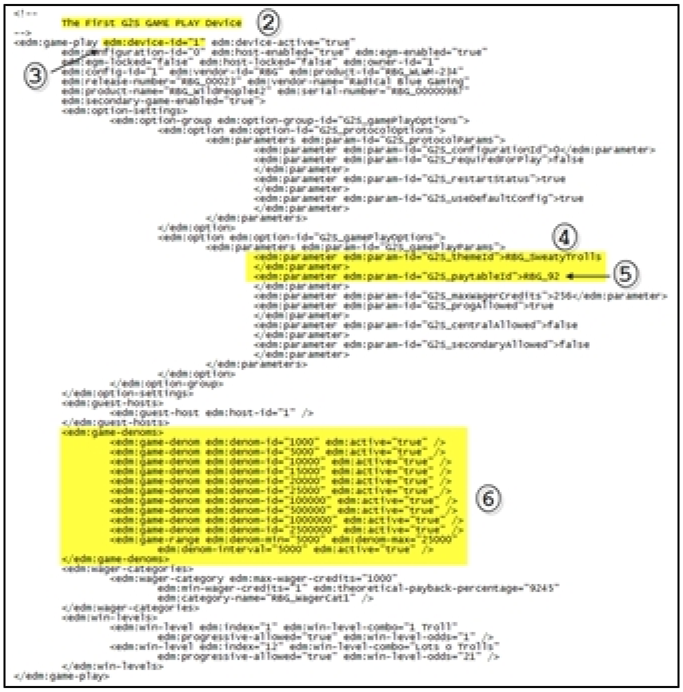
5. **Save** and **close** the file before loading it into RST.

Add 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.



```
<!--
      The First G2S GAME PLAY Device
-->
<edm:game-play 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:owner-id="1"
      edm:config-id="1" edm:vendor-id="RBG" edm:product-id="RBG_WLWM-234"
      edm:release-number="RBG_00023" edm:vendor-name="Radical Blue Gaming"
      edm:product-name="RBG_WildPeople42" edm:serial-number="RBG_00000987"
      edm:secondary-game-enabled="true">
      <edm:option-settings>
            <edm:option-group edm:option-group-id="G2S_gamePlayOptions">
                  <edm:option edm:option-id="G2S_protocolOptions">
                        <edm:parameters edm:param-id="G2S_protocolParams">
                              <edm:parameter edm:param-id="G2S_configurationId">0</edm:parameter>
                              <edm:parameter edm:param-id="G2S_requiredForPlay">false
                              </edm:parameter>
                              <edm:parameter edm:param-id="G2S_restartStatus">true
                              </edm:parameter>
                              <edm:parameter edm:param-id="G2S_useDefaultConfig">true
                              </edm:parameter>
                        </edm:parameters>
                  </edm:option>
                  <edm:option edm:option-id="G2S_gamePlayOptions">
                        <edm:parameters edm:param-id="G2S_gamePlayParams">
                              <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_maxWagerCredits">256</edm:parameter>
                              <edm:parameter edm:param-id="G2S_progAllowed">true
                              </edm:parameter>
                              <edm:parameter edm:param-id="G2S_centralAllowed">false
                              </edm:parameter>
                              <edm:parameter edm:param-id="G2S_secondaryAllowed">false
                              </edm:parameter>
                        </edm:parameters>
                  </edm:option>
            </edm:option-group>
      </edm:option-settings>
      <edm:guest-hosts>
            <edm:guest-host edm:host-id="1" />
      </edm:guest-hosts>
      <edm:game-denoms>
            <edm:game-denom edm:denom-id="1000" edm:active="true" />
            <edm:game-denom edm:denom-id="5000" edm:active="true" />
            <edm:game-denom edm:denom-id="10000" edm:active="true" />
            <edm:game-denom edm:denom-id="15000" edm:active="true" />
            <edm:game-denom edm:denom-id="20000" edm:active="true" />
            <edm:game-denom edm:denom-id="25000" edm:active="true" />
            <edm:game-denom edm:denom-id="100000" edm:active="true" />
            <edm:game-denom edm:denom-id="500000" edm:active="true" />
            <edm:game-denom edm:denom-id="1000000" edm:active="true" />
            <edm:game-denom edm:denom-id="2500000" edm:active="true" />
            <edm:game-range edm:denom-min="5000" edm:denom-max="25000"
                  edm:denom-interval="5000" edm:active="true" />
      </edm:game-denoms>
      <edm:wager-categories>
            <edm:wager-category edm:max-wager-credits="1000"
                  edm:min-wager-credits="1" edm:theoretical-payback-percentage="9245"
                  edm:category-name="RBG_WagerCat1" />
      </edm:wager-categories>
      <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>
</edm:game-play>
```

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.

Modify 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:

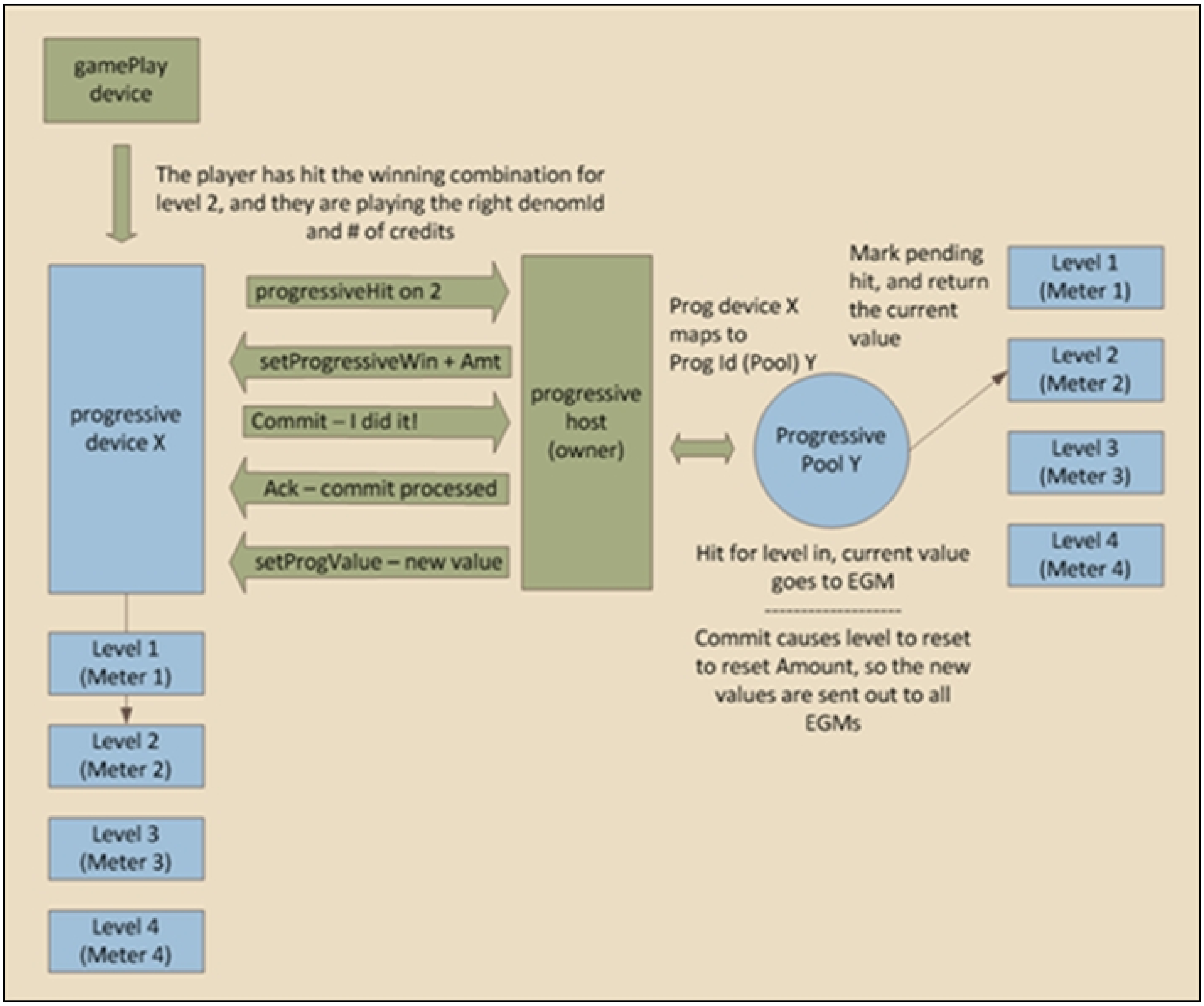
```
Progressive Device Section - Progressive Data Table
        <edm:option edm:option-id="G2S_progDataTable">
                <edm:parameters-table>
                        <edm:parameters edm:param-id="G2S_progData">
                        <edm:parameter edm:param-id="G2S_denomId">100000</edm:parameter>
                        <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.


gamePlay device
The player has hit the winning combination for level 2, and they are playing the right denomId and # of credits
progressive device X
Level 1 (Meter 1)
Level 2 (Meter 2)
Level 3 (Meter 3)
Level 4 (Meter 4)
progressiveHit on 2
setProgressiveWin + Amt
Commit – I did it!
Ack – commit processed
setProgValue – new value
progressive host (owner)
Prog device X maps to Prog Id (Pool) Y
Progressive Pool Y
Mark pending hit, and return the current value
Level 1 (Meter 1)
Level 2 (Meter 2)
Level 3 (Meter 3)
Level 4 (Meter 4)
Hit for level in, current value goes to EGM
Commit causes level to reset to reset Amount, so the new values are sent out to all EGMs

Configure Audit Meters

The `auditMeters` class enables the EGM to snapshot (at a specific time) and store a set of meters that may be requested by the host (through a `getMeterInfo` command) at any time prior to the next snapshot. At least one audit meter device must be active.

There are two options for a meters snapshot:

- **Minimum Set** - Supports class-level meters for all classes plus device-level meters for all active devices within the gamePlay class.
- **All Meters** - Supports all meters in the EGM.

Configure the smartegm-config File

1. Go to **RST-Install-Dir> smart-config >smart-egm> smartegm-config-gsa-am**.
2. Select the appropriate snapshot data:
3. Look for the `edm:minimal-audit-meters` attribute.
   - **false** (default) - Enables the Minimum Set option.
   - **true** - Enables All Meters option.

```
                                <!-- The Audit Meters Device -->

<edm:audit-meters xmlns:edm="http://www.radblue.com/egm-data-model/schemas/v1.0.0/"
edm:config-id="0" edm:configuration-id="0" edm:device-active="true" edm:device-id="1"
edm:egm-enabled="false" edm:egm-locked="false" edm:host-enabled="true"
edm:host-locked="false" edm:owner-id="1" edm:product-id="" edm:product-name="" edm:release-
number="" edm:serial-number="" edm:vendor-id="" edm:vendor-name="" edm:minimal-audit-
meters="false">
<edm:comment xmlns:edm="http://www.radblue.com/egm-data-model/schemas/v1.0.0/" />
<edm:guest-hosts xmlns:edm="http://www.radblue.com/egm-data-model/schemas/v1.0.0/" />
</edm:audit-meters>
```

Configure WAT Devices

There are two wat devices configured in the default smartegm-config-gsa.xml file. You can use this information to add additional wat devices or as a template for a custom smartegm-config file.

> **Note:** If the SmartEGM configuration file contains multiple WAT devices, a `G2S_WTX008 Unacknowledged Transaction in Log` error is generated for each WAT device *only*. Also, the entire WAT log is checked for unacknowledged transactions for each device rather than the last log entry.

In addition to the wat device, wat attributes can also be added to the voucher and handpay classes:

Class	WAT Attribute
handpay	G2S_disabledLocalWat
handpay	G2S_disabledRemoteWat
handpay	G2S_enabledLocalWat
voucher	G2S_titleWatPromo
voucher	G2S_titleWatNonCash

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>
```

Add 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	Unsupported events element. This

Name	Type	Use	Restrictions	Description
			maxOcc: 1	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, 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.

Reset 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**] > **script** > **smart-conf** > **smart-egm**

Right-click **smartegm-config.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 IGT Bonus Extensions

The IGT bonus class extensions add Multiple Jackpot Timer (MJT) and Wager Match bonusing capability. RST supports the following bonus class commands with IGT extensions:

Bonus Command	Direction
bonusAwardAck	RST-to-Host
bonusCommitStatus	RST-to-Host
bonusLogList	RST-to-Host
bonusProfile	RST-to-Host
bonusStatus	RST-to-Host
setBonusAward	Host-to-RST
setBonusAwardMulticast	Host-to-RST

If a `setBonusAward` is received during a game cycle, it is used for the game in progress.

The bonus award is always paid at `gameEnd`.

If the `setBonusAward` amount exceeds the *displayLimit*, the bonus is rejected (see *sbDirector Bonus.pdf* - section 3.8.4) and:

- the EGM sends a `commitBonus` with a *bonusException* equal to **9** (Bonus Limit Exceeded).
- the EGM generates a `G2S_BNE105` (Bonus Award Failed) event.

If the *displayLimit* is not zero, RST verifies that the amount of the award is less than or equal to *displayLimit*. If the amount of the award is greater than the *displayLimit*, RST fails the bonus award.

A new token, ***%x***, can be used in any of the bonus messages (this token was added to others in "Appendix E" of the G2S protocol document). If this token is present, it is replaced with the Total Wager Match escrow.

The following MJT autoplay features have **not** been implemented in RST:

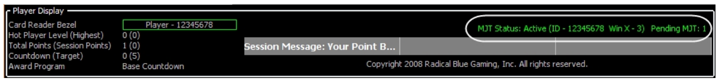
- *autoPlay*
- *timeoutRule*
- *timeout*
- *lowCreditReject*
- *cashOutEnd*
- *wagerRestriction*

IGT Bonus Extensions on the Player Display

When a bonus is awarded, the defined text message displays on the Player Display.

MJT Bonus Awards

RST supports multiple MJT award logic. When multiple MJT bonuses are sent to RST, the first MJT bonus becomes active. All subsequent bonuses are queued. The number of pending MJT bonuses can be seen on the Player Display.



Player Display

Card Reader Bezel	Player - 12345678
Hot Player Level (Highest)	0 (0)
Total Points (Session Points)	1 (0)
Countdown (Target)	0 (5)
Award Program	Base Countdown

MJT Status: Active (ID - 12345678 Win X - 3) Pending MJT: 1

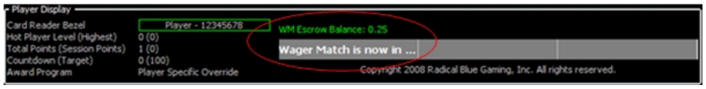
Session Message: Your Point B...

Copyright 2008 Radical Blue Gaming, Inc. All rights reserved.

Player Display showing queued wager match bonuses

Wager Match Awards

Wager Match Escrow Balance appears on the Player Display along with a "Wager Match is now in progress" message.



Player Display

Card Reader Bezel	Player - 12345678
Hot Player Level (Highest)	0 (0)
Total Points (Session Points)	1 (0)
Countdown (Target)	0 (100)
Award Program	Player Specific Override

WM Escrow Balance: 0.25

Wager Match is now in ...

Copyright 2008 Radical Blue Gaming, Inc. All rights reserved.

Player Display showing Wager Match bonus in-progress

If the EGM is unable to begin the bonus mode, the EGM sends a `commitBonus` command with *bonusPaidAmt* set to zero (0), with the appropriate *bonusException*. If the bonus mode is started, but the player never initiates a game and the bonus mode times out, the EGM returns exception code 0, Command successful.

If an error occurs, the text "Bonus Failed - Check Debug Log for Details" is shown on the Player Display.

When the bonus mode is canceled, the EGM sends the `commitBonus` command with *bonusPaidAmt* set to the amount won during the session and *bonusException* set to 1 (bonus canceled).

IGT Bonus Extensions in the SmartEGM Configuration File

The SmartEGM configuration file includes the IGT player extensions. The following IGT-specific attributes can be found in the player class section of the **smartegm-config-igt.xml** file.

Display Limit Player Option

- **IGT_ptDisplayLimit** - Maximum point balance the EGM displays.
- **IGT_ptOverLimitMsg** - Message to display when the maximum point balance is exceeded.
- **IGT_ptEarnedLimit** - Maximum number of points the EGM awards in single game cycle.
- **displayLimit** - Maximum bonus award (*bonusAwardAmt*) an EGM will accept. The *bonusAwardAmt* attribute is sent in the `setBonusAward` command.
- **displayLimitDuration** - Duration to display the *displayLimitText* message.
- **displayLimitText** - Text that is displayed if the bonus award exceeds the limit.
- **eligibleTimer** - Game is not eligible if it has been idle for this long (this functionality is controlled by the `setBonusAward` command).

Uncarded Player Option

- **IGT_uncardedSessionTimer** - Idle time required for EGM to terminate an uncarded player session.

Wager Match Option

- **IGT_wmActiveMsg** - Message to display in place of session message when a wager match authorization is active.
- **IGT_wmExitMsg** - Message to display when a Wager Match Authorization is closed.
- **IGT_wmDisplayLimit** - Total Wager Match Balance Limit.
- **IGT_wmOverLimitMsg** - Message to display when the sum of a wager match authorization exceeds the limit.
- **IGT_wmEarnPoints** - A player can earn points while a wager match authorization is active.

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.

Play Paytable Game

The Play Paytable Game verb simulates game play that uses paytables to report game outcomes. This verb is used primarily to test the Game Outcome extension of the G2S protocol. There are three fields specific to game outcome - Primary Outcome, Seed and Primary Outcome Count. You can automate game outcome testing through the Tiger scripting feature.

1. From the Player Verbs tab, click **Play Paytable Game**.
2. Configure the Play Paytable Game options as needed.
   - **Game Play Device ID -** Click the drop-down arrow, and select the game play device
   - **Theme -** The game theme type associated with the selected game play device. This field is *read-only*.
   - **Paytable ID -** The paytable identifier associated with the selected game play device.The information in this field is dependent on the Game Play Device ID value. This field is *read-only*. As the game is played the current paytable identifier of the EGM is updated to this value if it is not already set to it.
   - **Denom ID -** Click the drop-down arrow, and select the value of a single credit. The available denominations are from the game play device's active denom list.
   - **Cashable Wager -** Type or select the number of cashable, non-promotional credits you want to wager.
   - **Promo Wager -** Type or select the number of promotional credits you want to wager.
   - **Non-Cashable Wager -** Type or select the number of non-cashable, non-promotional credits you want to wager.
   - **Primary Outcome -** Click the drop-down arrow, and select the type of game outcome you want.

     If you are *not* using the outcome.xsd file, you can only select **random**, which indicates that RST randomly creates a game outcome.

     If you are using the outcome.xsd file, select a specific game outcome or select **random** to have a game outcome randomly created for you.
   - **Seed -** Type or select a number by which the random number generator is seeded. When used in conjunction with the outcome.xml file, specifying a seed number provides a consistent series of game outcomes.
   - **Primary Outcome Count -** Type or select, depending on the game type, the number of poker hands, keno games or reel lines to play for the primary game.

     Note that the primary outcome count must be a multiple of the total credits wagered.

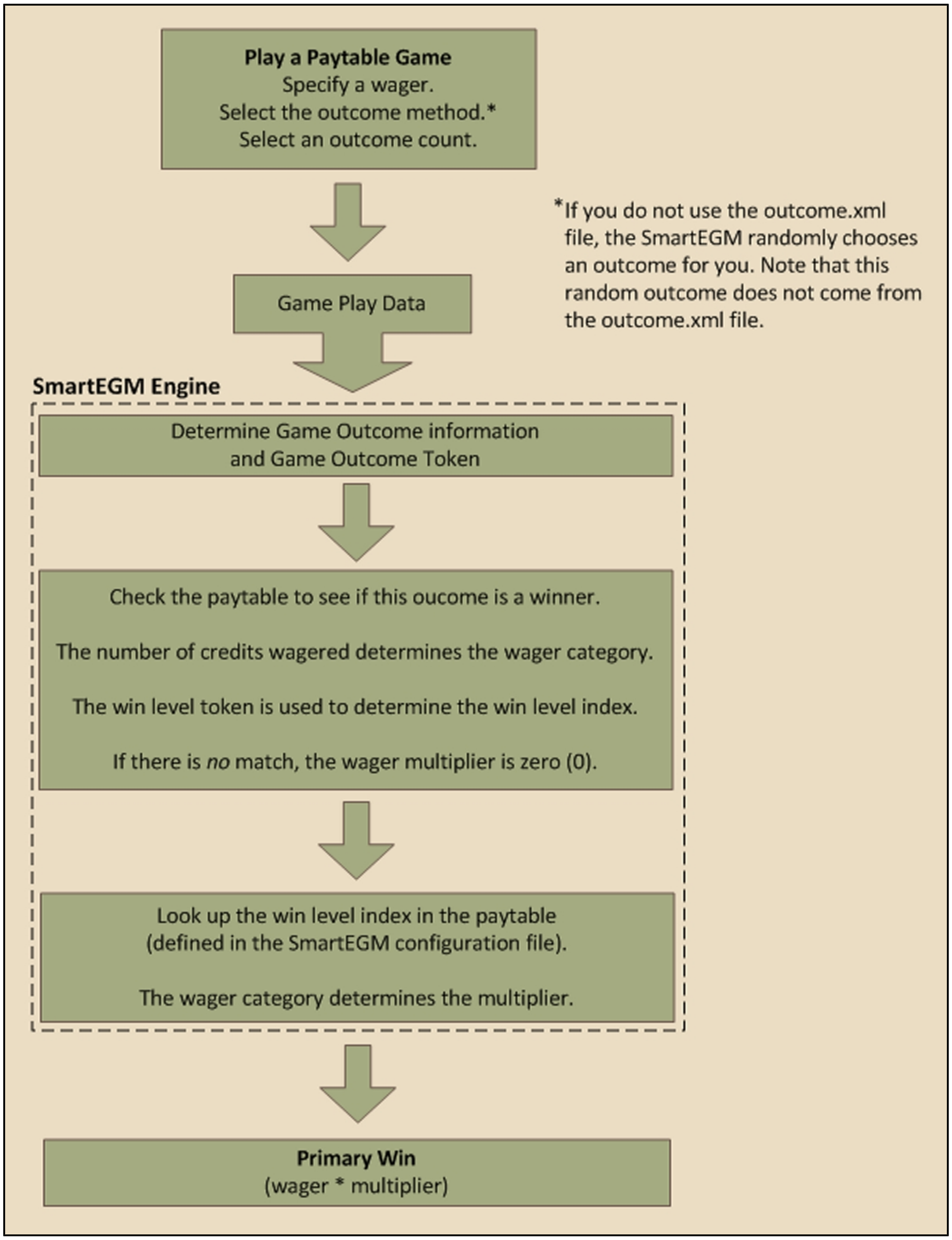
- **Secondary Game Count** - Type or select, how many secondary games to play. For each secondary game, the initial primary win is doubled.
- **Win on the final secondary game?** - If selected, the final secondary game is a winner and the final win of the game is the doubling of the initial win. If not selected, the final win of the game is zero (0).
- **Win to handpay?** - If selected, game play outcome is paid to a handpay, which is defined by the **How to pay handpay** field.
- **How to pay handpay** - Click the drop-down arrow, and select the type of handpay that should occur after game play has concluded.

  If you select Cancel a Handpay or Wait for Remote Key, the SmartEGM is locked and the winning credits must be keyed off (either through the Key Off Handpay player verb or by sending a `setRemoteKeyOff` command to RST, respectively).
- **Remote Key Off Time Out** - Type or select the number of milliseconds before the EGM times out when waiting to receive a `setRemoteKeyOff` command.
- **In Game Delay (milliseconds)** - Type or select the number of milliseconds to delay when a game completes.

3. Click **Play Game**.

How Game Outcome Works

When a paytable game is played - either through the Play Paytable Game player verb or 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.


Play a Paytable Game
Specify a wager.
Select the outcome method.*
Select an outcome count.
*If you do not use the outcome.xml file, the SmartEGM randomly chooses an outcome for you. Note that this random outcome does not come from the outcome.xml file.
Game Play Data
SmartEGM Engine
Determine Game Outcome information and Game Outcome Token
Check the paytable to see if this oucome is a winner.
The number of credits wagered determines the wager category.
The win level token is used to determine the win level index.
If there is no match, the wager multiplier is zero (0).
Look up the win level index in the paytable (defined in the SmartEGM configuration file).
The wager category determines the multiplier.
Primary Win
(wager * multiplier)

When using the Play Paytable Game player verb with a customized outcome.xml file, the Primary Outcome field displays all outcome names defined in the outcome.xml file. When a game is played, the SmartEGM randomly selects a game outcome for the selected game type from the outcome.xml file.

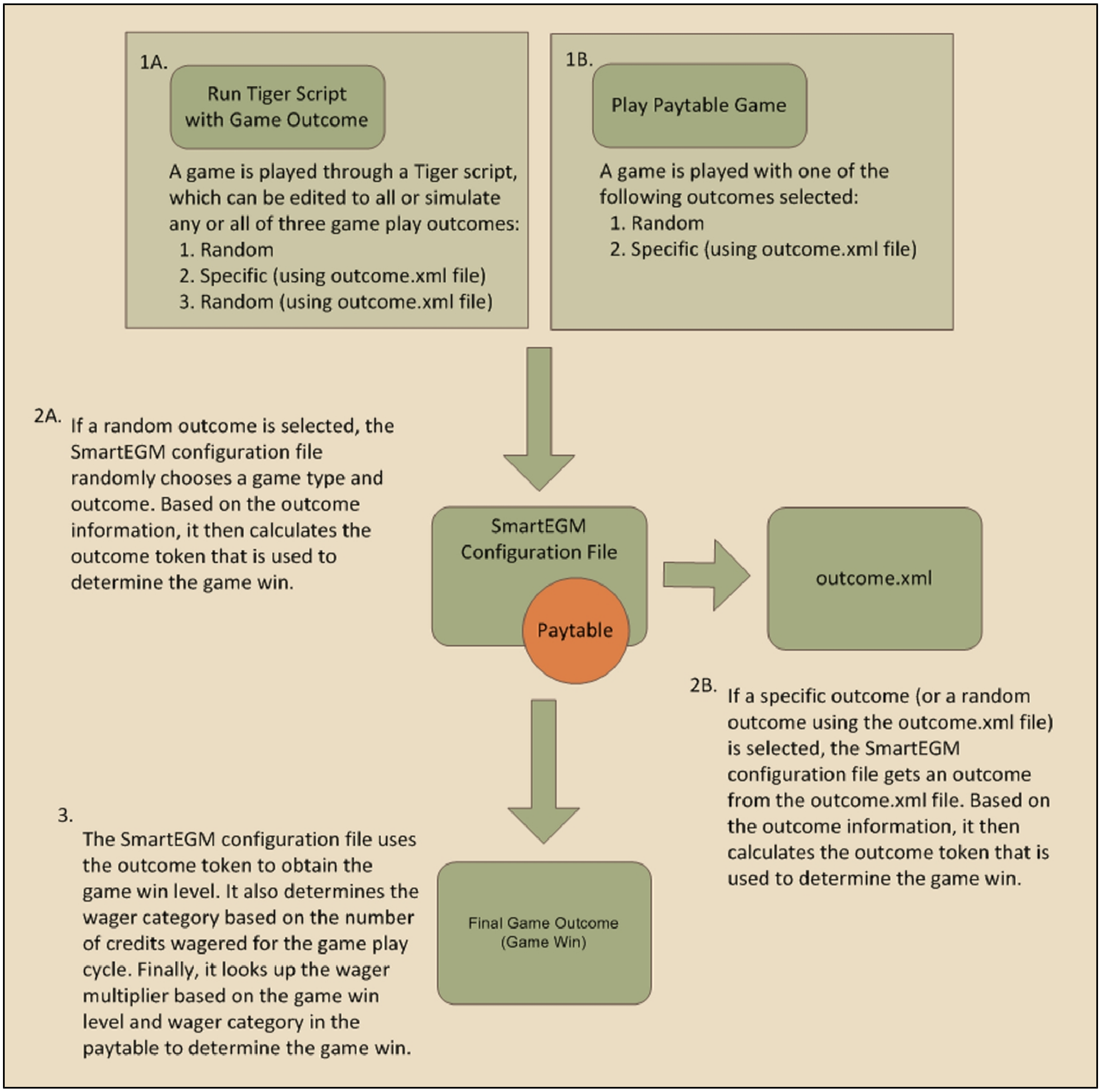
The final game outcome is sent in the `IGT_GPE103` (Final Outcome Logged) event, which you can view through the Message Transcript.

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.

	Random	**Random from File**	**Specific from File**
User Interface		X	X
Tiger Verb	X	X	X

Game Outcome Flow in RST


1A.
Run Tiger Script with Game Outcome
1B.
Play Paytable Game
SmartEGM Configuration File
Paytable
outcome.xml
Final Game Outcome (Game Win)

1A. A game is played through a Tiger script, which can be edited to all or simulate any or all of three game play outcomes:

1. Random
2. Specific (using outcome.xml file)
3. Random (using outcome.xml file)

1B. A game is played with one of the following outcomes selected:

1. Random
2. Specific (using outcome.xml file)

2A. If a random outcome is selected, the SmartEGM configuration file randomly chooses a game type and outcome. Based on the outcome information, it then calculates the outcome token that is used to determine the game win.

2B. If a specific outcome (or a random outcome using the outcome.xml file) is selected, the SmartEGM configuration file gets an outcome from the outcome.xml file. Based on the outcome information, it then calculates the outcome token that is used to determine the game win.

3. The SmartEGM configuration file uses the outcome token to obtain the game win level. It also determines the wager category based on the number of credits wagered for the game play cycle. Finally, it looks up the wager multiplier based on the game win level and wager category in the paytable to determine the game win.

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 overview in RST:

1. Configure paytable information in the SmartEGM configuration file.
2. Define game outcomes (*optional*).
3. Configure Tiger scripting for automated testing.
   or
4. Use the Play Paytable Game player verb to send manual game outcome messages.

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**] > **script** > **smart-conf** > **smart-egm**
2. Right-click **smartegm-config.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.
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
paytable-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.

When you edit the file, the new content will be used the next time RST is started.

To modify the outcome.xml file:

1. Navigate to the **RST installation directory**, and open the **conf** folder.
2. Open the **EGM-1** file.
3. Right-click the **outcome.xml** file, select **Open With…** > **Notepad**.
4. Use the outcome.xml elements and attributes as well as the game outcome example to modify the outcome.xml file.
5. **Save** and **close** the file.
6. Restart RST.
   a. From the RST **SmartEGM** layout, click the **Main** subtab.
   b. Click **Stop SmartEGM**.
   c. Click **Start SmartEGM** to re-start RST.

   The modified game outcome content will be used when RST starts.

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	The name of the outcome record.
weight	type: positive integer use: optional default: 1	The weight of this outcome record when randomly 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 the final 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

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.

About Schema Validation

By default, the tool validates all received G2S commands. There may be times, however, especially during the testing process, that you want the tool to reject or accept specific commands - whether they are valid or not - *or* disable schema validation completely.

Schema validation management is useful when you want to ignore specific commands, but continue with testing, or when you want to do regression testing.

To modify how specific commands are handled, you must first save the sample G2S validation manager XML file, which resides in the tool's installation directory, as a new **g2s-validation-manager.xml** file. This allows the file to be saved when the tool is uninstalled.

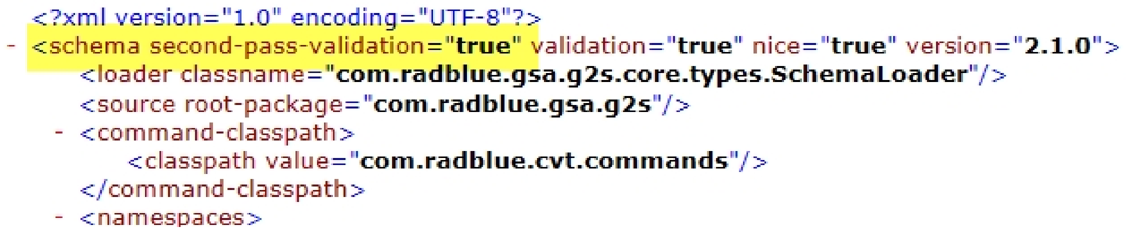
You can also disable validation completely. You may want to disable validation if:

- validation is affecting the tool's performance by using too much CPU.
- your EGM or host is in violation, but you cannot correct the issues immediately.

Second Validation

This attribute is designed to give you a second validation pass. G2S validator attribute (**second-pass-validate)** has been added to the **[install directory] >schemas > g2s > 2.1.0 >g2s-schema-metadata.xml**.

- True - (Default) - A G2S message undergoes a second, validation pass-designed to detect syntax errors that are not found in the first validation pass attribute.
- False - If the second validation pass takes too much CPU resources, or the EGM/host is in violation of the G2S schema and you are unable to fix your code.



```
  <?xml version="1.0" encoding="UTF-8"?>
- <schema second-pass-validation="true" validation="true" nice="true" version="2.1.0">
    <loader classname="com.radblue.gsa.g2s.core.types.SchemaLoader"/>
    <source root-package="com.radblue.gsa.g2s"/>
  - <command-classpath>
      <classpath value="com.radblue.cvt.commands"/>
    </command-classpath>
  - <namespaces>
```

Customize Schema Validation

When you customize the command validation for the tool’s current schema, you are defining exceptions to the tool’s default of validating all commands. To define these exceptions, you must edit the validation manager file, and save the file without the “-sample” extension into the schema directory used by the tool.

1. Go to: [**installation directory**] > **schemas** > **g2s** > [**schema directory**]
2. Right-click the **g2s-validation-manager-sample.xml** file, and select **Edit**.
3. Go to: **File** > **Save As**, and navigate to the directory of the schema you’ll being using with the tool.
4. Change the file name to **g2s-validation-manager.xml**, and click **Save**.
5. Modify the command name, and add new command name attributes, as needed.

   In the G2S validation manager file, there are two attributes:

   - **command name**

     The class and command name of the command you want to affect, using the notation: [class].[command]

     To affect all commands in a specific class, use an asterisk (*) to denote all commands. For example, the line

     ```
     <command name=download.* state=“ACCEPT” />
     ```

     would instruct the tool to accept all download commands without validating them.

   - **state**

     The state attribute lets you define the action you want the tool to take for the specified command(s). Since the tool, by default, validates all commands, you want to define the exceptions. Your choices are:

     - **Accept** - The tool accepts the specified command without validating it. When a command is accepted without validation, the tool returns a `g2sAck` response command. Note that, if the command is not valid, the tool will take no further action.
     - **Reject** - The tool rejects the specified command without validating it. When a command is rejected, the tool sends an `MSX004` (Incomplete/Malformed XML) error in the `g2sAck` response command.
6. Select **File**, and click **Save** to save your changes.
7. If the tool is running, stop it.
8. Start the tool. The tool now responds to the commands specified in the validation manager file as defined.

Disable Schema Validation

You can disable schema validation by changing an attribute in the **g2s-schema-metadata.xml** file.

1. Go to: [**installation directory**] > **schemas** > **g2s** > [**schema directory**]
2. Right-click the **g2s-schema-metadata.xml** file, and select **Edit**.
3. Change the *validation* attribute to **false** to disable validation.
4. Select **File**, and click **Save** to save your changes.
5. If the tool is running, stop it.
6. Start the tool. The tool no longer validates G2S commands as they are received.

Working with the Message Transcript

The Message Transcript lets you examine individual commands sent from, or received by, the tool. The data displayed is extracted directly from received G2S or S2S messages (depending on the protocol you are using).

Filtering options offer you a variety of ways to view information. Each instance of the transcript within the tool can be filtered differently.

At the top of the Transcript screen are several options:

- **Host Selector** - Click the drop-down arrow, and select to view the transcript information for a specific host system or all host systems.
- **Load** - Load transcript messages from the database so you can work with them through the user interface.
- **Compare** - View the details of any two messages side-by-side.
- **Filters** - This control allows you to select which commands are to be included on the display. Make any changes, and then press **OK** to have the tool update the display. This control currently resets when a new data set is loaded.
- **Search Content** - Search through the contents of all displayed messages in this transcript instance for the entered text pattern (case sensitive). Clicking on a row in the returned list gives you access to the HTTP header and message contents of the selected message.
- **Set Comment** - Adds a comment to the Comment column of the selected message.
- **Clear Display** - Clears the displayed messages in this instance of the transcript control.
- **Clear Database** - Clears all records of this type in the database for this instance of the tool.
- **Analyze** - *Available for G2S messaging only*. Provides a user-friendly summary of transcript messages.

Transcript Column Headers

The following columns are available in the transcript:

- **Command ID** - Command ID associated with message.
- **Comment** - Information entered by the user about a specific message. This field is *not* part of the actual message. Comments exist *only* in the tool in which they are entered.
- **Date Received** - Date and time message was received by the tool.

- **Device** - Class and identifier of device where the message is being sent to or from.
- **Event Code** - Code for associated event.
- **Event Date/Time** - Date and time that event was sent.
- **Event ID** - Identifier for associated event.
- **Event Text** - Description of associated event.
- **From Location** - Identifier of entity (for example, EGM or host) that sent the message.
- **Message ID** - Unique identifier associated with the message.
- **Session ID** - Session ID associated with message.
- **Session Type** - Indicates how the message should be processed: as a request, response or notification.
- **Summary** - Actual G2S or S2S command within the message. If more than one command is sent in a message, only the first command appears in the transcript. However, all commands with the message are displayed in the detail view, which you can access by double-clicking the message.

  For `G2S_GPE112 (Game Ended)` events, the outcome of game play (the *playResult* attribute) is shown in the Summary column in brackets (for example, **eventReport:G2S_GPE112 - Game Ended [Lost]**).
- **To Location** - Identifier of the intended target of the message.

You can slide the columns around to rearrange their order. To move a column header, left-click and hold while you move the column to it's new location. You can also click any column to re-sort it, or use CTRL + left-click to sort on multiple columns.

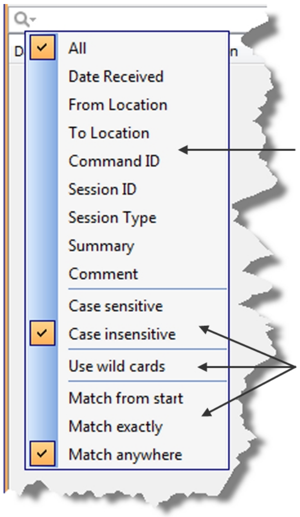
If you right-click on a column header, a menu displays that allows you to automatically resize one or all columns (based on the displayed data in the columns), as well as to indicate which columns you would like to display.

Clicking on any column header causes the data to be sorted using that header. Click once to sort the column in ascending order. Click a again to sort the column in descending order. The third click clears the sort.

If you want to sort on multiple columns, use the CTRL key when clicking the column headers.

Filter Transcript Messages Using the Quick Filter

Just below the Transcript options is a magnifying glass and entry field that allows you to filter messages based on entered data. To filter the displayed data, click inside the entry field and start typing. The displayed data is automatically filtered as you type.



Select the fields you want to search.

Select the search criteria.

Clicking on the magnifying glass gives you a menu that you can use to provide additional selection criteria.

This powerful tool allows you to immediately view any set of messages that you can imagine, limited only by the data displayed in the columns.

What Are You Looking for in the Transcript?

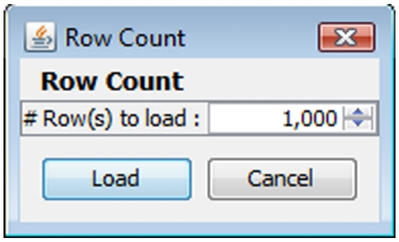
- *Are the correct commands being sent?*
  For example, is the `commsOnLine` command being sent during startup?
- *Are messages being acknowledged* (for example, with a `g2sAck` as well as a message acknowledgements?
- *Are there correct request-response pairs?*
  For example, if a `communications.getDescriptor` command is sent (outbound), a corresponding `communications.descriptorList` command should be received (inbound). Note that the Session ID is the same for the request and response.

Load Messages into the Transcript

The Load option lets you display a pre-defined number of messages in the transcript.

If you are using RGS, be sure the EGM for which you want to view information is selected.

1. Click the **Transcript** tab on the **Transcripts** layout.
2. From the Transcript object, click **Load**.



Row Count

Row Count

Row(s) to load : 1,000

3. Type the number of messages you want to display, and click **Load**.
   The Transcript display populates with the requested number of messages.

Compare Messages in the Transcript

The Compare option lets you view the details of two messages, side-by-side. You can choose to view the message content in three different formats: in a user-friendly format, XML format, and XML format with the differences between the two messages highlighted in **red**.

The Compare option is available when you view the Event Report as well as in the main Message Transcript.

1. While holding down the CTRL key, click the two messages you want to compare.
2. Click **Compare**.

   The selected messages display side-by-side, allowing you to scroll through the details of each message to compare them.

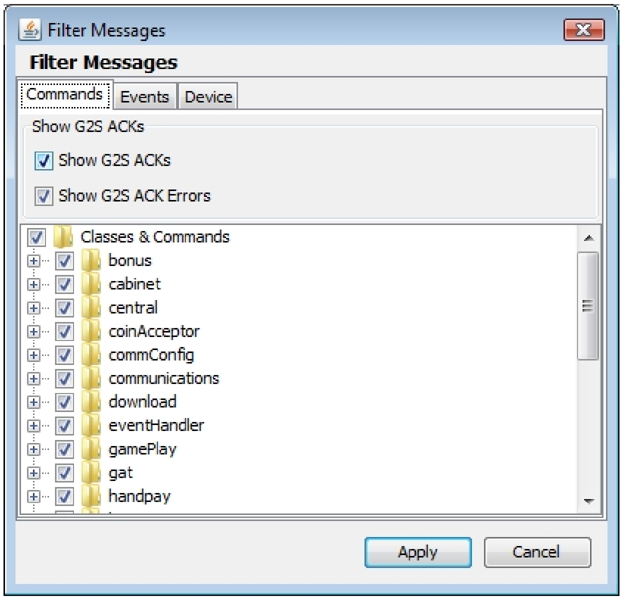
   The **Command** tab displays command information in an easy-to-read format along with any meters.

4. Click **View XML** to view the XML for the messages.
5. Click **Diff** to view the XML with the changes highlighted in red.
6. Click **OK** to close the compare view.

Filter Messages in the Transcript

The Filters option in the Transcript lets you select the commands, events or devices you want to display in the transcript window. Use this option to narrow the transcript view to just the messages that interest you. The excluded data is not deleted from the transcript database; it is just not displayed and can always be included again.

1. Click the **Transcript** tab on the **Transcripts** layout.
2. Click **Filters**. The Filter Messages screen displays with three tabs: Commands, Events and Device.



Show G2S ACKs

☑ Show G2S ACKs

☑ Show G2S ACK Errors

☑ Classes & Commands
- ☑ bonus
- ☑ cabinet
- ☑ central
- ☑ coinAcceptor
- ☑ commConfig
- ☑ communications
- ☑ download
- ☑ eventHandler
- ☑ gamePlay
- ☑ gat
- ☑ handpay

3. On each tab, select the check box of the commands, events and devices you want to display in the Transcript, and clear the check box of the commands, events and devices you want to hide.

   By default, the `g2sAck` command is cleared (does not display).

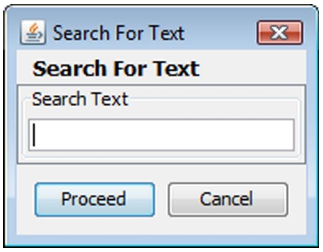
4. Click **OK**.

   Your changes take effect immediately.

Search the Content of Transcript Messages

The Search Content option lets you search transcript message content for keywords.

1. Click the **Transcript** tab on the **Transcripts** layout.
2. Click **Search Content**.



3. Type the keywords for your search, and click **Proceed**.
   The **Transcript Search Results** screen displays all conforming messages.
4. To view the details of a message, double-click the message.
5. Click **Back** to close the Transcript Search Results window.

View Command Objects through the Transcript

A command object is a graphical representation of a command (as opposed to viewing the command in XML format). You can view command objects on the message details screen. List information for complex commands displays in tabs to the right of the command attributes. From the message details screen you can link to the `g2sAck` message, corresponding request-response pair command or to the associated SOAP message for any command. Attributes that are deprecated in G2S 2.1 display in a strike-through font (for example, ~~Enable Money Out~~). Comments associated with the message are displayed next to the Command and View XML tabs.

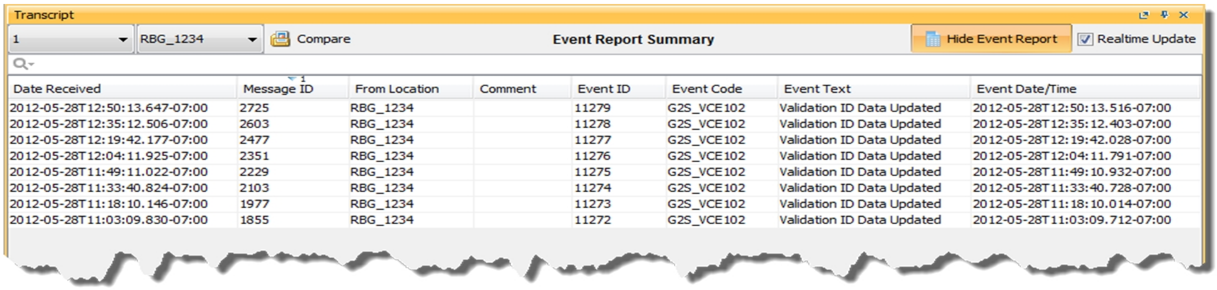
To display a message's command object:

1. Double-click the command you want to view.
2. Click the **Command** tab.
3. Click **Previous** and **Next** to navigate through the transcript list while in the message detail view.
4. Click **g2sAck** to view the corresponding acknowledgment to the selected command.
5. Click **Req-Resp** to view the corresponding command in the request-response pair.
6. Click **Transport Message** to view the command's associated SOAP message.
7. Click **OK** to return to the transcript.

View the Event Report

The Event Report displays all events sent and received by the tool. You can toggle between the Transcript view and the event view by clicking **Show/Hide Event Report**. The Compare option lets you compare two event messages.

1. Click the **Transcript** tab on the **Transcripts** layout.
2. Click **Show Event Report**.



Date Received	Message ID	From Location	Comment	Event ID	Event Code	Event Text	Event Date/Time
2012-05-28T12:50:13.647-07:00	2725	RBG_1234		11279	G2S_VCE102	Validation ID Data Updated	2012-05-28T12:50:13.516-07:00
2012-05-28T12:35:12.506-07:00	2603	RBG_1234		11278	G2S_VCE102	Validation ID Data Updated	2012-05-28T12:35:12.403-07:00
2012-05-28T12:19:42.177-07:00	2477	RBG_1234		11277	G2S_VCE102	Validation ID Data Updated	2012-05-28T12:19:42.028-07:00
2012-05-28T12:04:11.925-07:00	2351	RBG_1234		11276	G2S_VCE102	Validation ID Data Updated	2012-05-28T12:04:11.791-07:00
2012-05-28T11:49:11.022-07:00	2229	RBG_1234		11275	G2S_VCE102	Validation ID Data Updated	2012-05-28T11:49:10.932-07:00
2012-05-28T11:33:40.824-07:00	2103	RBG_1234		11274	G2S_VCE102	Validation ID Data Updated	2012-05-28T11:33:40.728-07:00
2012-05-28T11:18:10.146-07:00	1977	RBG_1234		11273	G2S_VCE102	Validation ID Data Updated	2012-05-28T11:18:10.014-07:00
2012-05-28T11:03:09.830-07:00	1855	RBG_1234		11272	G2S_VCE102	Validation ID Data Updated	2012-05-28T11:03:09.712-07:00

   - **Date Received** - Date and time message was received by the tool.
   - **Event Code** - Code associated with event (for example, `G2S_VCE102`).
   - **Event Date/Time** - Date and time message was received by the tool.
   - **Event ID** - Event identifier.
   - **Event Text** - Text of associated event code (for example, Validation ID Data Updated).
   - **From Location** - Identifier of entity (for example, EGM or host) that sent the event.
   - **Message ID** - Message identifier.

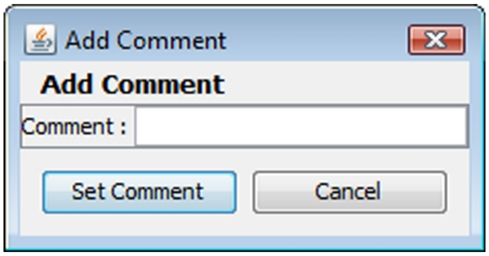
3. Double-click any record to view the details of the message in which the event was sent.
4. Click **OK** to close the message detail screen.
5. Click **Hide Event Report** to return to the Transcript view.

Add a Comment to a Transcript Message

The Set Comment option lets you add a comment to any message in the transcript. Comments are part of the transcript only. Messages are not modified by comments.

> **Note:** Comments associated with a message are displayed next to the Command and View XML tabs when you view the message details.

1. Single-click the message to highlight it.
2. Click **Set Comment**.



3. Type your comment, and click **Set Comment**.

   The message is highlighted in blue, and the comment appears in the Comment field.

Export Message Content to Excel

The Export to Excel File right-click option lets you quickly export the content of a G2S command to a Microsoft Excel spreadsheet. The following G2S commands can be exported to Excel:

- **`communications.descriptorList`**
  Exports all attributes for each supported device, sorted by device class and device ID.
- **`eventHandler.eventSubList`**
- **`eventHandler.setEventSub`**
- **`eventHandler.supportedEvents`**
  If an event in the `eventHandler.supportedEvents` command does not contain the event text, the event text is added automatically if it is not included in the original message.
- **`meters.meterInfo`**
- **`optionConfig.optionList`**

To export message content to an Excel spreadsheet:

1. Click the **Transcript** tab on the **Transcripts** layout.
2. Click to highlight the command you want to export.
3. Right-click inside the selected record, and select **Export to Excel File**.
4. Type a name for the file.
5. Click **Save**.
   A confirmation message displays with the number of records saved in the file.
6. Click **OK**.

Export Transcript Entries

The Export to Transcript File option lets you quickly export all or select Transcript entries for review. This file can then be imported into the RadBlue Analysis Suite (RAS) for further analysis.

This option is useful when an issue has been narrowed down to a limited set of commands. Rather than exporting a large set of troubleshooting data using the Export Debug option on the Debug tab, you can send only the specific command(s) involved in the issue.

You can import the Transcript file directly into RAS by using the Import option.

1. Click the **Transcript** tab on the **Transcripts** layout.
2. Select the Transcript records you want to export.

   There are several ways to select records:

   a. Click to select a single record.
   b. CTRL+click to select multiple records.
   c. SHIFT+click to select a block of records. Click once on the first record and once on the last record to highlight the entire block.
   d. For another way to select a block of records, right-click and hold on the first row. Then, drag the cursor down the screen, highlighting each row as you go, until you come to the last row you want to export.
   e. CTRL+a to select all Transcript records.
3. Right-click inside the Transcript, and select **Export to Transcript File**.
4. Type a name for the file.
5. Click the Files of Type drop-down arrow, and select either a **S2S Transcript (*.s2t)** or a **G2S Transcript (*.g2)** file extension. Your selection corresponds to the protocol you are using with the tool (either G2S or S2S).
6. Click **Save**.

   A confirmation message displays with the number of records saved in the file.
7. Click **OK**.

Clear the Transcript Display

The Clear Display option lets you clear all transcript messages from the table. This option *does not* remove messages from the transcript database.

Click **Clear > Display** to remove all messages from the current view.

Clear the Transcript Database

The Clear Database option lets you remove all transcript records from the transcript database, clears transcript messages from the table and clears the EGM selector list. Note that this action cannot be undone.

1. Click **Clear > Database**.
2. Click **Yes** to remove all data from the transcript database, or click **No** to return to the transcript without clearing the database.

About the EGM Transcript Analysis Report

The EGM Transcript Analysis report provides information about messages that were sent from and received by the application for the period requested. The purpose of this report is to provide the user with a user-friendly summary of G2S messages.

The report is divided into several sections to assist your analysis of the information:

- **Transcript Summary** - Information related to the computer running the installed application.
- **Transcript** - Transaction log of sent and received G2S and S2S messages. G2S acknowledgements and, optionally, keepAlive messages are filtered out.
- **Sessions** - Transcript messages grouped by session ID.
- **Device Commands** - Transcript messages grouped by affected G2S device.
- **Device States** - Status of each device and any device status changes for the requested time period.
- **Events** - Events generated by the EGM.
- **G2S ACK Errors** - G2S acknowledgement messages containing errors.
- **Meters** - Meter values.
- **Messages** - G2S XML messages from the transcript.

The EGM Transcript Analysis report can be output to an HTML page or to an XML file.

Generate the Transcript Analysis Report

1. Click **Analyze**. If you are using the Advanced Transcript Analyzer feature, click the **Analyze** drop-down arrow and select **General**.
   - **EGM ID** (required) - *RGS only*. Type the identifier of the EGM on which you want to report. If you are using a RadBlue product other than RGS, this field does not display.
   - **Start Date** - Click the drop-down arrow to select the beginning date and time of the reporting period.
   - **End Date** - Click the drop-down arrow to select the ending date and time of the reporting period.
   - **Ignore keepAlives** - Select if you do not want to see `keepAlive` and `keepAliveAck` commands on your report.
   - **OutPut Options** - Select **Produce Transcript Report** to export the Transcript Analysis report directly to HTML, or select **Produce Transcript File** to export the Transcript Analysis report to an XML file. If you export the report to an XML file, you can accept the default output location or browse to the location of your choice.
2. Click **Start Analysis Process**.

Navigating the Transcript Analysis Report File

The top of the report contains links to each section.

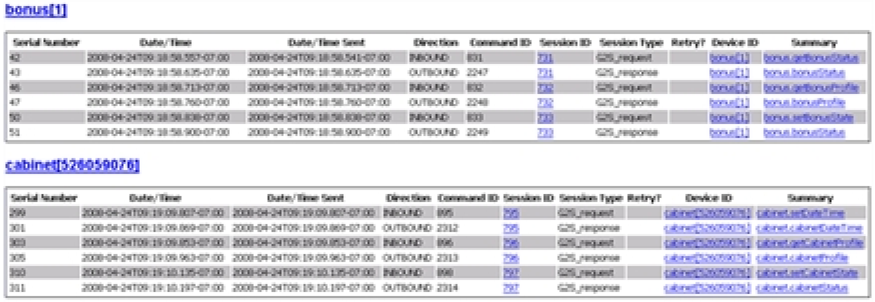
- Click a link to move through the file.
- Click the browser's **Back** button to return to the previous location in the file.

Sample Transcript Report

Device Commands

The Device Commands section contains sent and received transcript messages grouped by individual G2S devices.

If Session ID numbers are reused, each row will contain all of the messages with the same Session ID. Currently, the report does not display legal pairs.



bonus[1]

Serial Number	Date/Time	Date/Time Sent	Direction	Command ID	Session ID	Session Type	Retry?	Device ID	Summary
42	2008-04-24T09:18:58.557-07:00	2008-04-24T09:18:58.541-07:00	INBOUND	831	731	G2S_request		bonus[1]	bonus.getBonusStatus
43	2008-04-24T09:18:58.635-07:00	2008-04-24T09:18:58.635-07:00	OUTBOUND	2247	731	G2S_response		bonus[1]	bonus.bonusStatus
46	2008-04-24T09:18:58.713-07:00	2008-04-24T09:18:58.713-07:00	INBOUND	832	732	G2S_request		bonus[1]	bonus.getBonusProfile
47	2008-04-24T09:18:58.760-07:00	2008-04-24T09:18:58.760-07:00	OUTBOUND	2248	732	G2S_response		bonus[1]	bonus.bonusProfile
50	2008-04-24T09:18:58.838-07:00	2008-04-24T09:18:58.838-07:00	INBOUND	833	733	G2S_request		bonus[1]	bonus.setBonusState
51	2008-04-24T09:18:58.900-07:00	2008-04-24T09:18:58.900-07:00	OUTBOUND	2249	733	G2S_response		bonus[1]	bonus.bonusStatus

cabinet[526059076]

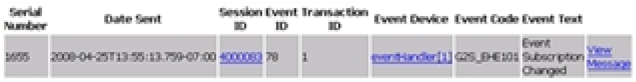
Serial Number	Date/Time	Date/Time Sent	Direction	Command ID	Session ID	Session Type	Retry?	Device ID	Summary
299	2008-04-24T09:19:09.807-07:00	2008-04-24T09:19:09.807-07:00	INBOUND	895	795	G2S_request		cabinet[526059076]	cabinet.setDateTime
301	2008-04-24T09:19:09.869-07:00	2008-04-24T09:19:09.869-07:00	OUTBOUND	2312	795	G2S_response		cabinet[526059076]	cabinet.cabinetDateTime
303	2008-04-24T09:19:09.853-07:00	2008-04-24T09:19:09.853-07:00	INBOUND	896	796	G2S_request		cabinet[526059076]	cabinet.getCabinetProfile
305	2008-04-24T09:19:09.963-07:00	2008-04-24T09:19:09.963-07:00	OUTBOUND	2313	796	G2S_response		cabinet[526059076]	cabinet.cabinetProfile
310	2008-04-24T09:19:10.135-07:00	2008-04-24T09:19:10.135-07:00	INBOUND	898	797	G2S_request		cabinet[526059076]	cabinet.setCabinetState
311	2008-04-24T09:19:10.197-07:00	2008-04-24T09:19:10.197-07:00	OUTBOUND	2314	797	G2S_response		cabinet[526059076]	cabinet.cabinetStatus

Device States

Status of each device and any device status changes for the requested time period.

Events

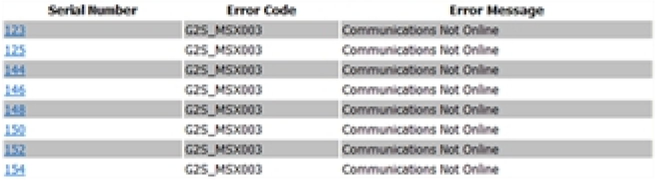
The Events section displays information about events generated by the EGM.



Serial Number	Date Sent	Session ID	Event ID	Transaction ID	Event Device	Event Code	Event Text	
1655	2008-04-25T13:55:13.759-07:00	4000083	78	1	eventHandler[1]	G2S_EHE101	Event Subscription Changed	View Message

G2S ACKs That Have Errors

The G2S ACKs That Have Errors section shows G2S acknowledgement messages containing errors.

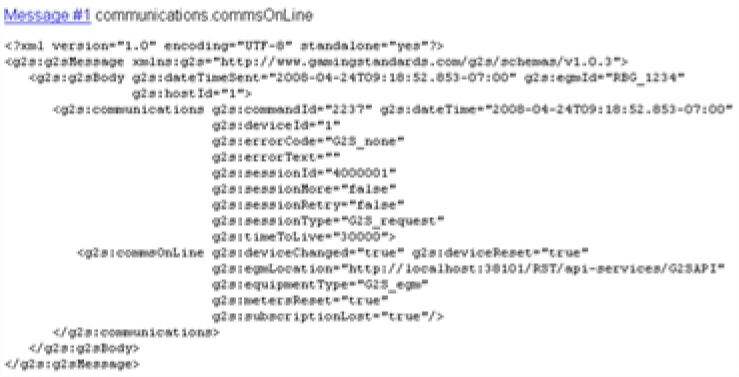


Serial Number	Error Code	Error Message
123	G2S_MSX003	Communications Not Online
125	G2S_MSX003	Communications Not Online
144	G2S_MSX003	Communications Not Online
146	G2S_MSX003	Communications Not Online
148	G2S_MSX003	Communications Not Online
150	G2S_MSX003	Communications Not Online
152	G2S_MSX003	Communications Not Online
154	G2S_MSX003	Communications Not Online

Messages

The Messages section displays the actual XML code for messages that appear in the transcript.

The report may rewrite the XML content while formatting this section of the report. In particular, the report may change the XML namespace prefix. However, the XML namespace URI is maintained. Note that unused namespace declarations may not be displayed.



Message #1 communications.commsOnLine

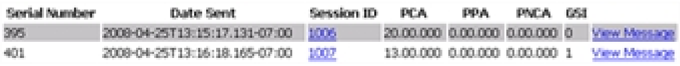
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<g2s:g2sMessage xmlns:g2s="http://www.gamingstandards.com/g2s/schemas/v1.0.3">
   <g2s:g2sBody g2s:dateTimeSent="2008-04-24T09:18:52.853-07:00" g2s:egmId="RBG_1234"
                g2s:hostId="1">
      <g2s:communications g2s:commandId="2237" g2s:dateTime="2008-04-24T09:18:52.853-07:00"
                          g2s:deviceId="1"
                          g2s:errorCode="G2S_none"
                          g2s:errorText=""
                          g2s:sessionId="4000001"
                          g2s:sessionMore="false"
                          g2s:sessionRetry="false"
                          g2s:sessionType="G2S_request"
                          g2s:timeToLive="30000">
         <g2s:commsOnLine g2s:deviceChanged="true" g2s:deviceReset="true"
                          g2s:egmLocation="http://localhost:38101/RST/api-services/G2SAPI"
                          g2s:equipmentType="G2S_egm"
                          g2s:metersReset="true"
                          g2s:subscriptionLost="true"/>
      </g2s:communications>
   </g2s:g2sBody>
</g2s:g2sMessage>
```

Meters

The Meters section shows the values of the following meters:

- PCA - Player Cashable Amount (G2S_playerCashableAmt)
- PPA - Player Promo Amount (G2S_playerPromoAmt)
- PNCA - Player Non-Cashable Amount (G2S_playerNonCashAmt)
- GSI - Games Since Initialization (G2S_gamesSinceInitCnt)

Note that this is not a comprehensive list of all possible meters.



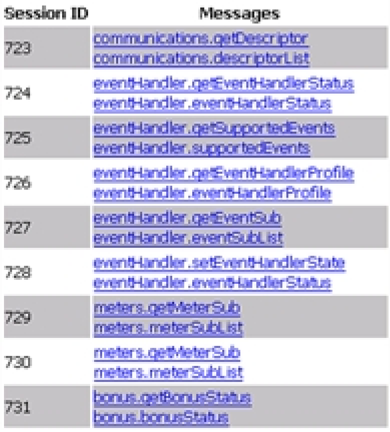
Serial Number	Date Sent	Session ID	PCA	PPA	PNCA	GSI	
395	2008-04-25T13:15:17.131-07:00	1006	20.00.000	0.00.000	0.00.000	0	View Message
401	2008-04-25T13:16:18.165-07:00	1007	13.00.000	0.00.000	0.00.000	1	View Message

Sessions

The Sessions section groups sent and received messages. Messages appear in ascending order, by **Session ID**.

A Session ID is a number that is set by the sender of a request, which allows the response to be tied to the original request. While session ID numbers should increase, they don't have to - they must be unique for the number of outstanding messages.

Entries in red indicate incomplete sessions (either no response or too many responses) and are often errors.

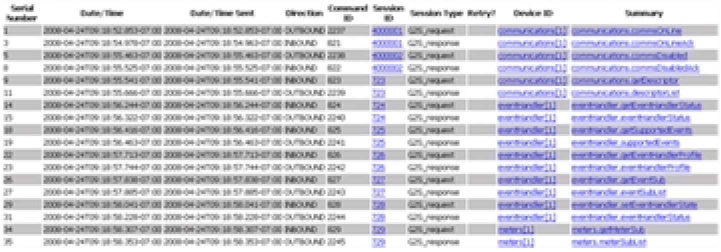


Session ID	Messages
723	communications.getDescriptor communications.descriptorList
724	eventHandler.getEventHandlerStatus eventHandler.eventHandlerStatus
725	eventHandler.getSupportedEvents eventHandler.supportedEvents
726	eventHandler.getEventHandlerProfile eventHandler.eventHandlerProfile
727	eventHandler.getEventSub eventHandler.eventSubList
728	eventHandler.setEventHandlerState eventHandler.eventHandlerStatus
729	meters.getMeterSub meters.meterSubList
730	meters.getMeterSub meters.meterSubList
731	bonus.getBonusStatus bonus.bonusStatus

Transcript

The Transcript section shows sent and received messages. Note that since the date is from the G2S message (sent either by the EGM or the host), the date may be incorrect. There are a number of reasons why this occurs. For example:

- the PC clock is incorrect.
- the Network Time Protocol (NTP) is not functioning correctly.
- there are programming errors.



Serial Number	Date/Time	Date/Time Sent	Direction	Command ID	Session ID	Session Type	Retry?	Device ID	Summary
1	2008-04-24T09:18:52.853-07:00	2008-04-24T09:18:52.853-07:00	OUTBOUND	2237	4000001	G2S_request		communications[1]	communications.commsOnLine
3	2008-04-24T09:18:54.978-07:00	2008-04-24T09:18:54.963-07:00	INBOUND	621	4000001	G2S_response		communications[1]	communications.commsOnLineAck
5	2008-04-24T09:18:55.463-07:00	2008-04-24T09:18:55.463-07:00	OUTBOUND	2238	4000002	G2S_request		communications[1]	communications.commsDisabled
8	2008-04-24T09:18:55.525-07:00	2008-04-24T09:18:55.525-07:00	INBOUND	622	4000002	G2S_response		communications[1]	communications.commsDisabledAck
9	2008-04-24T09:18:55.541-07:00	2008-04-24T09:18:55.541-07:00	INBOUND	623	723	G2S_request		communications[1]	communications.getDescriptor
11	2008-04-24T09:18:55.666-07:00	2008-04-24T09:18:55.666-07:00	OUTBOUND	2239	723	G2S_response		communications[1]	communications.descriptorList
14	2008-04-24T09:18:56.244-07:00	2008-04-24T09:18:56.244-07:00	INBOUND	624	724	G2S_request		eventHandler[1]	eventHandler.getEventHandlerStatus
15	2008-04-24T09:18:56.322-07:00	2008-04-24T09:18:56.322-07:00	OUTBOUND	2240	724	G2S_response		eventHandler[1]	eventHandler.eventHandlerStatus
18	2008-04-24T09:18:56.416-07:00	2008-04-24T09:18:56.416-07:00	INBOUND	625	725	G2S_request		eventHandler[1]	eventHandler.getSupportedEvents
19	2008-04-24T09:18:56.463-07:00	2008-04-24T09:18:56.463-07:00	OUTBOUND	2241	725	G2S_response		eventHandler[1]	eventHandler.supportedEvents
22	2008-04-24T09:18:57.713-07:00	2008-04-24T09:18:57.713-07:00	INBOUND	626	726	G2S_request		eventHandler[1]	eventHandler.getEventHandlerProfile
23	2008-04-24T09:18:57.744-07:00	2008-04-24T09:18:57.744-07:00	OUTBOUND	2242	726	G2S_response		eventHandler[1]	eventHandler.eventHandlerProfile
26	2008-04-24T09:18:57.838-07:00	2008-04-24T09:18:57.838-07:00	INBOUND	627	727	G2S_request		eventHandler[1]	eventHandler.getEventSub
27	2008-04-24T09:18:57.885-07:00	2008-04-24T09:18:57.885-07:00	OUTBOUND	2243	727	G2S_response		eventHandler[1]	eventHandler.eventSubList
29	2008-04-24T09:18:58.041-07:00	2008-04-24T09:18:58.041-07:00	INBOUND	628	728	G2S_request		eventHandler[1]	eventHandler.setEventHandlerState
31	2008-04-24T09:18:58.228-07:00	2008-04-24T09:18:58.228-07:00	OUTBOUND	2244	728	G2S_response		eventHandler[1]	eventHandler.eventHandlerStatus
34	2008-04-24T09:18:58.307-07:00	2008-04-24T09:18:58.307-07:00	INBOUND	629	729	G2S_request		meters[1]	meters.getMeterSub
35	2008-04-24T09:18:58.353-07:00	2008-04-24T09:18:58.353-07:00	OUTBOUND	2245	729	G2S_response		meters[1]	meters.meterSubList

The **Serial Number** is assigned by the tool. This number corresponds to the order in which the message was received by the application. Use this value to compare the arrival order between any two messages.

The **Date/Time Sent** field is the date and time message that is sent to the host or EGM.

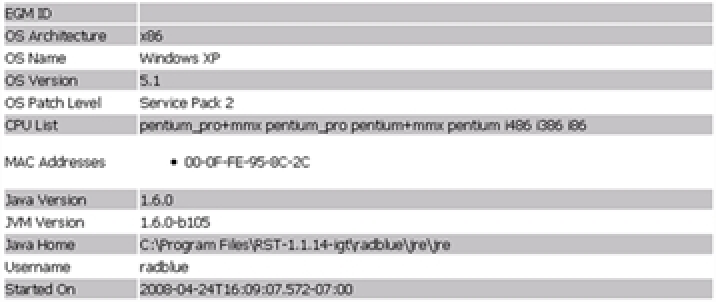
The **Date/Time** field is the date and time message was constructed.

The **Direction** is relative to the application. *Inbound* means a message that came from the other end. *Outbound* is a message that was generated by the application you ran the report from.

The **Retry?** column is blank for first attempts and filled in for retries. This column corresponds to the *sessionRetry* attribute in the G2S message.

Transcript Summary

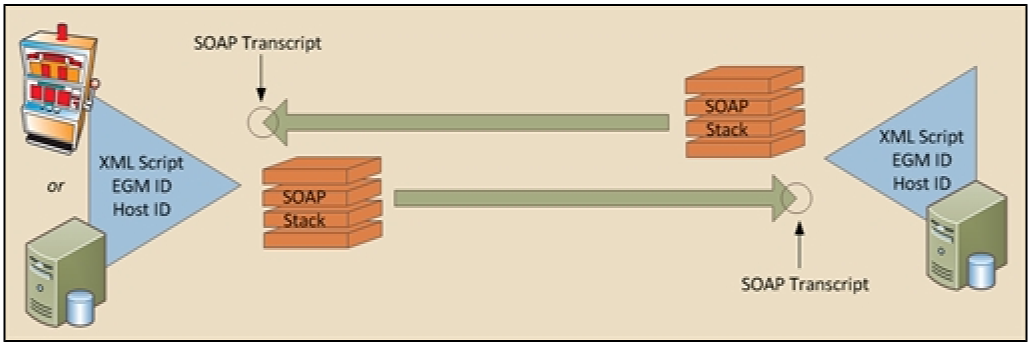
The Transcript Summary section shows information related to the computer running the installed application.



EGM ID	
OS Architecture	x86
OS Name	Windows XP
OS Version	5.1
OS Patch Level	Service Pack 2
CPU List	pentium_pro+mmx pentium_pro pentium+mmx pentium i486 i386 i86
MAC Addresses	• 00-0F-FE-95-8C-2C
Java Version	1.6.0
JVM Version	1.6.0-b105
Java Home	C:\Program Files\RST-1.1.14-igt\radblue\jre\jre
Username	radblue
Started On	2008-04-24T16:09:07.572-07:00

About SOAP Messages

Simple Object Access Protocol (SOAP) is an XML-based protocol for transporting information in a decentralized, distributed environment. A SOAP message consists of a mandatory SOAP envelope, an optional SOAP header, and a mandatory SOAP body. G2S or S2S message content is contained within the body.



SOAP message flow.

- **Envelope** - Top element of the XML document representing the message.
- **Header** - Generic mechanism for adding features to a SOAP message in a decentralized manner without prior agreement between the communicating parties. SOAP defines a few attributes that can be used to indicate who should deal with a feature and whether it is optional or mandatory.
- **Body** - Container for mandatory information intended for the ultimate recipient of the message. SOAP defines one element for the body, which is the Fault element used for reporting errors.

The SOAP Transcript lets you view SOAP-encapsulated messages as they come off the SOAP stack.

Working with the SOAP Transcript

The SOAP Transcript lets you view inbound SOAP-encapsulated messages as they come off the SOAP stack. Here, you can see details of the SOAP wrapper around the messages plus the messages themselves. Primarily, this transcript shows you how the endpoint (for example, an EGM or host) constructed the SOAP around the message. It is used to debug connection issues at startup. SOAP faults display in the SOAP Transcript and are noted in the Summary column as *SOAP Exceptions*.

The SOAP Transcript has several options that allow you to view SOAP information. Use the detail view to see the SOAP wrapper and the message it contains.

At the top of the SOAP Transcript screen are several options:

- **Search Content** - Search through the contents of all displayed messages in this transcript instance for the entered text pattern (case sensitive). Clicking on a row in the returned list gives you access to the HTTP header and message contents of the selected message. See Search the Content of a SOAP Message.
- **Clear Display** - Clears the displayed messages in this instance of the Transcript. See Clear the SOAP Transcript Display.
- **Clear DB** - Clears all records of this type in the database for this instance of the tool. See Clear the SOAP Transcript Database.

The size limit of the SOAP Transcript is 4MB. If this limit is reached, messages beyond the limit are not stored in the transcript database, and an informational message displays in the debug log.

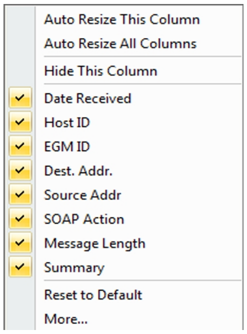
For more information on SOAP messages, see About SOAP Messages.

SOAP Transcript Column Headers

The following columns are available in the SOAP Transcript:

- **Date Received** - Date and time message was received by the tool.
- **Host ID** - Unique identifier of the host.
- **EGM ID** - Unique identifier of the EGM.
- **Dest. Addr.** - Destination Address. URL of the application that the message is sent to.
- **Source Addr** - Source Address. URL of the application that the message was sent from.
- **SOAP Action** - HTTP header value from the SOAP message. If you are having an issue with the SOAP connection, look at the SOAP Action column and verify that the values are consistent with GSA guidelines.
- **Message Length** - Number of bytes in the SOAP message.
- **Summary** - Actual G2S or S2S command within the SOAP message. If more than one command is sent in a message, only the first command appears in the transcript. However, all commands with the message are displayed in the detail view, which you can access by double-clicking the message.

You can slide the columns around to rearrange their order. To move a column header, left-click and hold while you move the column to it's new location. You can also click any column to re-sort it, or use CTRL + left-click to sort on multiple columns.



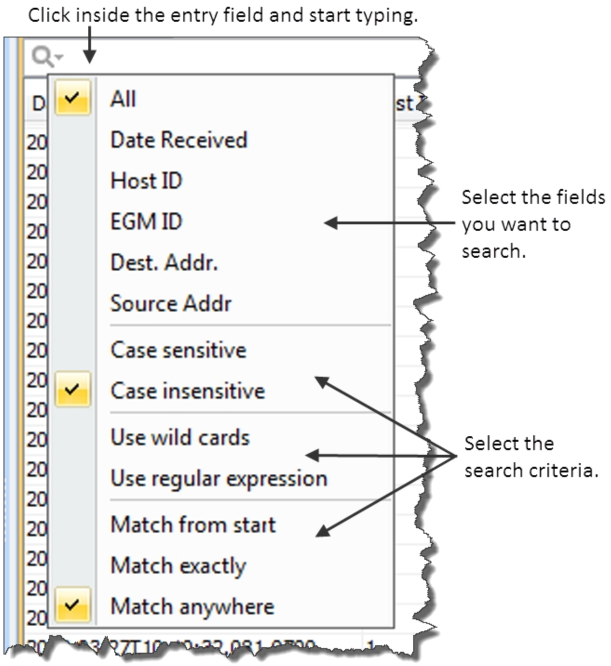
If you right-click on a column header, a menu displays that allows you to automatically resize one or all columns (based on the displayed data in the columns), as well as to indicate which columns you would like to display.

Clicking on any column header causes the data to be sorted using that header. Click once to sort the column in ascending order. Click a again to sort the column in descending order. The third click clears the sort.

If you want to sort on multiple columns, use the CTRL key when clicking the column headers.

Filter SOAP Transcript Messages Using the Quick Filter

Just below the SOAP Transcript options is a magnifying glass and entry field that allows you to filter messages based on entered data. To filter the displayed data, click inside the entry field and start typing. The displayed data is automatically filtered as you type.



Clicking on the magnifying glass gives you a menu that you can use to provide additional selection criteria.

This powerful tool allows you to immediately view any set of messages that you can imagine, limited only by the data displayed in the columns.

What Are You Looking for in the SOAP Transcript?

The SOAP Transcript lets you view SOAP-encapsulated messages, as they come off the SOAP stack. Primarily, this transcript shows you how the EGM, Central server or Edge server constructs the SOAP around the message. Use the detail view to see the SOAP wrapper and the message it contains.

Receive Real-Time Data in the SOAP Transcript

If you select Realtime Update, the screen updates dynamically as messages are processed by the tool. However, real-time updating may cause application slowdown.

View the Content of a SOAP Message

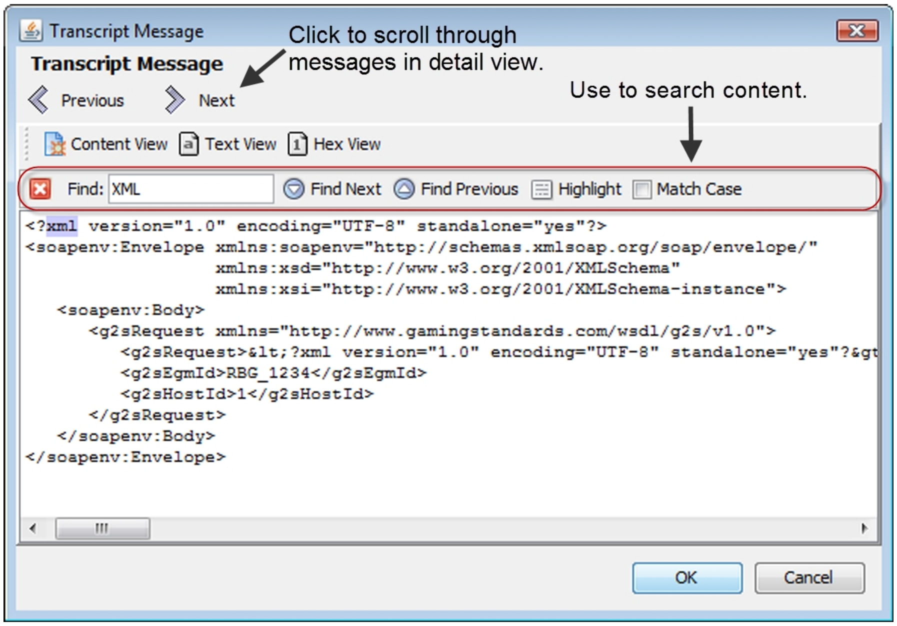
By double-clicking on any row, you can examine the details of the SOAP message. The SOAP message detail view lets you switch between a text view of the SOAP wrapper, a hexadecimal view of the SOAP wrapper and the message's XML content. You can also browse through the SOAP transcript list while in detail view.

The Find option lets you search for text strings and keywords within the message. Simply click in the Find text box and start typing. The tool instantly jumps to the first match of the entered string. This feature becomes very handy when you want to find data in large XML messages.

- 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 message.
- Select **Match Case** if you want to find only a text string or keyword with a specific case (capital or lower case letters).

To view the content of a SOAP message:

1. Double-click the message you want to view.
2. Click **Text View** to see the SOAP message details as text.



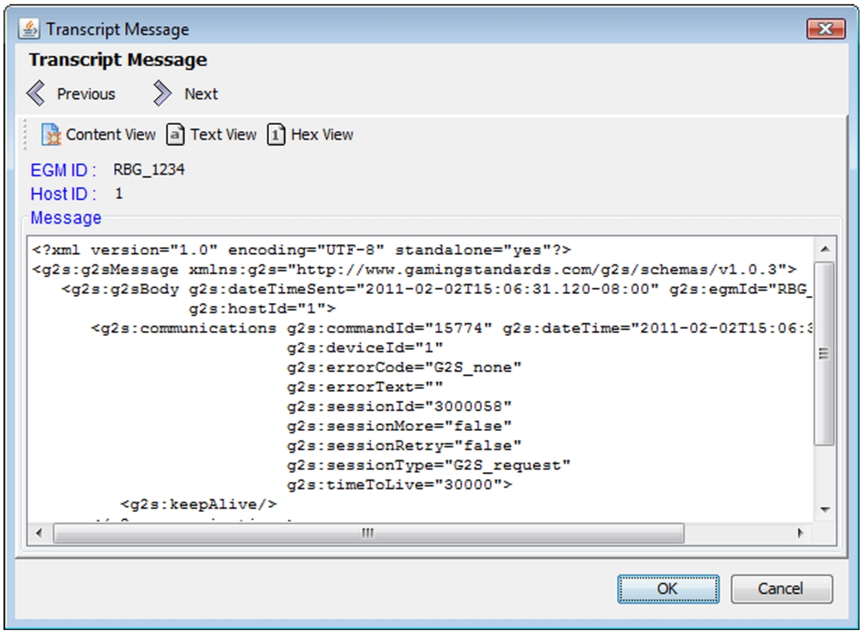
Click to scroll through messages in detail view.

Use to search content.

Find: XML

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/"
                  xmlns:xsd="http://www.w3.org/2001/XMLSchema"
                  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
   <soapenv:Body>
      <g2sRequest xmlns="http://www.gamingstandards.com/wsdl/g2s/v1.0">
         <g2sRequest>&lt;?xml version="1.0" encoding="UTF-8" standalone="yes"?&gt
         <g2sEgmId>RBG_1234</g2sEgmId>
         <g2sHostId>1</g2sHostId>
      </g2sRequest>
   </soapenv:Body>
</soapenv:Envelope>
```

3. Click **Content View** to view the message content.



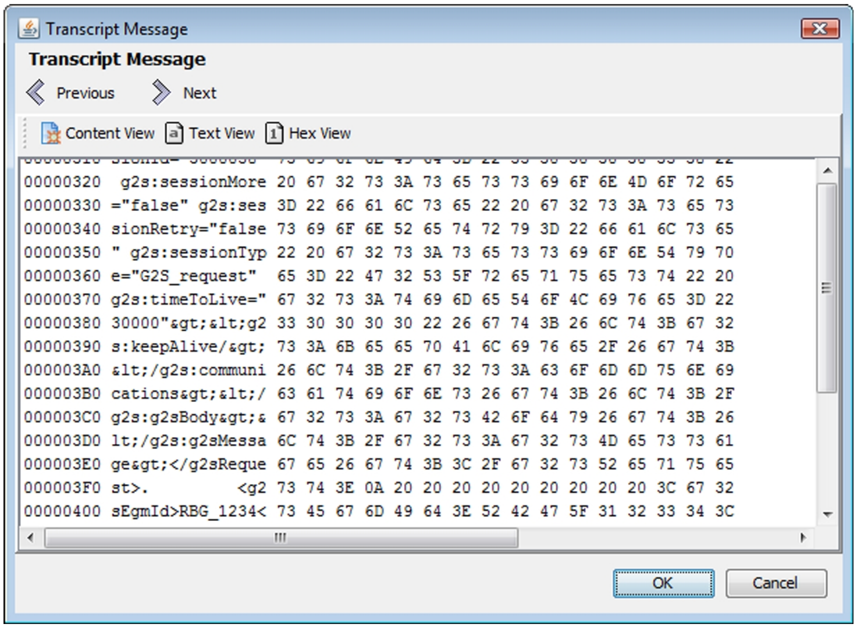
EGM ID : RBG_1234

Host ID : 1

Message

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<g2s:g2sMessage xmlns:g2s="http://www.gamingstandards.com/g2s/schemas/v1.0.3">
   <g2s:g2sBody g2s:dateTimeSent="2011-02-02T15:06:31.120-08:00" g2s:egmId="RBG_
                g2s:hostId="1">
      <g2s:communications g2s:commandId="15774" g2s:dateTime="2011-02-02T15:06:3
                          g2s:deviceId="1"
                          g2s:errorCode="G2S_none"
                          g2s:errorText=""
                          g2s:sessionId="3000058"
                          g2s:sessionMore="false"
                          g2s:sessionRetry="false"
                          g2s:sessionType="G2S_request"
                          g2s:timeToLive="30000">
         <g2s:keepAlive/>
```

4. Click **Hex View** to see the SOAP message details as hexadecimal.



```
00000320  g2s:sessionMore 20 67 32 73 3A 73 65 73 73 69 6F 6E 4D 6F 72 65
00000330 ="false" g2s:ses 3D 22 66 61 6C 73 65 22 20 67 32 73 3A 73 65 73
00000340 sionRetry="false 73 69 6F 6E 52 65 74 72 79 3D 22 66 61 6C 73 65
00000350 " g2s:sessionTyp 22 20 67 32 73 3A 73 65 73 73 69 6F 6E 54 79 70
00000360 e="G2S_request"  65 3D 22 47 32 53 5F 72 65 71 75 65 73 74 22 20
00000370 g2s:timeToLive=" 67 32 73 3A 74 69 6D 65 54 6F 4C 69 76 65 3D 22
00000380 30000"&gt;&lt;g2 33 30 30 30 30 22 26 67 74 3B 26 6C 74 3B 67 32
00000390 s:keepAlive/&gt; 73 3A 6B 65 65 70 41 6C 69 76 65 2F 26 67 74 3B
000003A0 &lt;/g2s:communi 26 6C 74 3B 2F 67 32 73 3A 63 6F 6D 6D 75 6E 69
000003B0 cations&gt;&lt;/ 63 61 74 69 6F 6E 73 26 67 74 3B 26 6C 74 3B 2F
000003C0 g2s:g2sBody&gt;& 67 32 73 3A 67 32 73 42 6F 64 79 26 67 74 3B 26
000003D0 lt;/g2s:g2sMessa 6C 74 3B 2F 67 32 73 3A 67 32 73 4D 65 73 73 61
000003E0 ge&gt;</g2sReque 67 65 26 67 74 3B 3C 2F 67 32 73 52 65 71 75 65
000003F0 st>.         <g2 73 74 3E 0A 20 20 20 20 20 20 20 20 20 3C 67 32
00000400 sEgmId>RBG_1234< 73 45 67 6D 49 64 3E 52 42 47 5F 31 32 33 34 3C
```

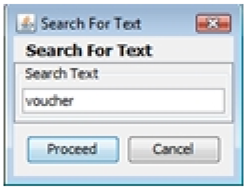
5. Click **OK** to return to the SOAP Transcript. Click **Previous** and **Next** to navigate through the SOAP transcript list while in message detail view.

6. Click **OK** to return to the SOAP transcript.

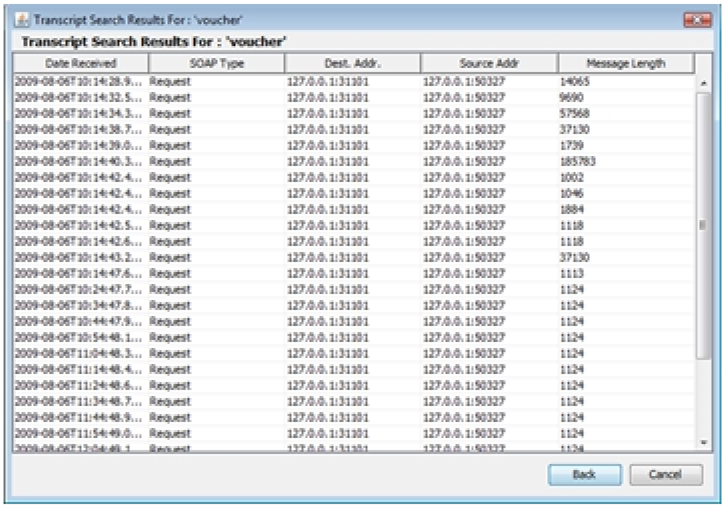
Search the Content of a SOAP Message

The Search Content option lets you search for keywords within all messages currently displayed in the SOAP transcript.

1. Type the information you want to search on.



2. Click **Proceed**, or click **Cancel** to return to the SOAP Transcript. A pop-up window displays all messages containing the text you entered.



Transcript Search Results For : 'voucher'

Date Received	SOAP Type	Dest. Addr.	Source Addr	Message Length
2009-08-06T10:14:28.9...	Request	127.0.0.1:31101	127.0.0.1:50327	14065
2009-08-06T10:14:32.5...	Request	127.0.0.1:31101	127.0.0.1:50327	9690
2009-08-06T10:14:34.3...	Request	127.0.0.1:31101	127.0.0.1:50327	57568
2009-08-06T10:14:38.7...	Request	127.0.0.1:31101	127.0.0.1:50327	37130
2009-08-06T10:14:39.0...	Request	127.0.0.1:31101	127.0.0.1:50327	1739
2009-08-06T10:14:40.3...	Request	127.0.0.1:31101	127.0.0.1:50327	185783
2009-08-06T10:14:42.4...	Request	127.0.0.1:31101	127.0.0.1:50327	1002
2009-08-06T10:14:42.4...	Request	127.0.0.1:31101	127.0.0.1:50327	1046
2009-08-06T10:14:42.4...	Request	127.0.0.1:31101	127.0.0.1:50327	1884
2009-08-06T10:14:42.5...	Request	127.0.0.1:31101	127.0.0.1:50327	1118
2009-08-06T10:14:42.6...	Request	127.0.0.1:31101	127.0.0.1:50327	1118
2009-08-06T10:14:43.2...	Request	127.0.0.1:31101	127.0.0.1:50327	37130
2009-08-06T10:14:47.6...	Request	127.0.0.1:31101	127.0.0.1:50327	1113
2009-08-06T10:24:47.7...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T10:34:47.8...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T10:44:47.9...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T10:54:48.1...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:04:48.3...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:14:48.4...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:24:48.6...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:34:48.7...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:44:48.9...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T11:54:49.0...	Request	127.0.0.1:31101	127.0.0.1:50327	1124
2009-08-06T12:04:49.1...	Request	127.0.0.1:31101	127.0.0.1:50327	1124

3. Click any message to display the SOAP envelope along with the XML message text.
4. Click **Back** to return to the SOAP Transcript.

Clear the SOAP Transcript Display

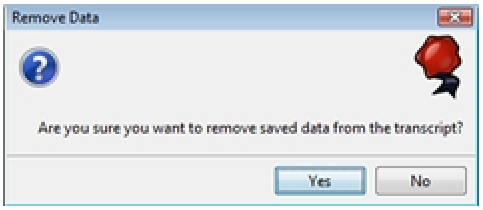
The Clear Display option lets you clear all messages from the SOAP transcript. This option does not remove messages from the SOAP Transcript database.

Click **Clear Display** to remove all messages from the current view.

Clear the SOAP Transcript Database

The Clear DB option lets you remove all messages from the SOAP Transcript database. Note that this action cannot be undone.

1. Click **Clear DB**.


Remove Data

Are you sure you want to remove saved data from the transcript?

2. Click **Yes** to remove all data from the SOAP Transcript database, or click **No** to return to the Transcript without clearing the database.

Working with the Multicast Transcript

The Multicast Transcript displays multicast messages that are sent between the host and EGM. Note that G2S commands contained in multicast messages can also be viewed through the Transcript. However, the Transcript does not display multicast wrapper information.

At the top of the Multicast Transcript Control screen are several options:

- **Search Content** - Search through the contents of all displayed messages in this transcript instance for the entered text pattern (case sensitive). Clicking on a row in the returned list gives you access to the HTTP header and message contents of the selected message. See Search the Content of a Multicast Message.
- **Clear Display** - Clears the displayed messages in this instance of the transcript control. See Clear the Multicast Transcript Display.
- **Clear DB** - Clears all records of this type in the database for this instance of the tool. See Clear the Multicast Transcript Database.
- **Clear Multicast Listeners** - *RPA only*. Clears all endpoint multicast listeners, which are persisted in RPA. See Clear the Multicast Listeners.

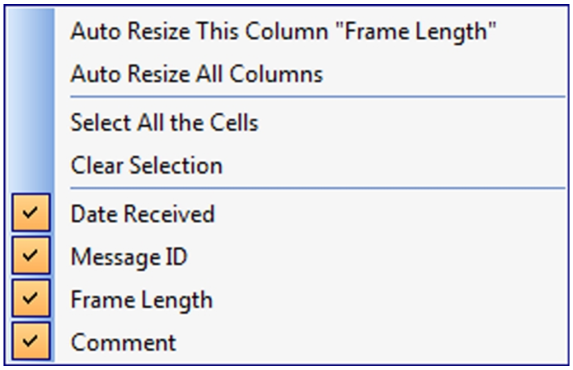
The size limit of the multicast transcript is 4MB. If this limit is reached, messages beyond the limit are not stored in the transcript database, and an informational message displays in the debug log.

Multicast Transcript Column Headers

The following columns are available in the Multicast Transcript:

- **Date Received** - Date and time message is received by the tool.
- **Message ID** - Identification number of the message.
- **Frame Length** - Length of the frame contents.
- **Comment** - Indicates an invalid UMAC (Message Authentication Code using Universal Hashing) or an invalid frame length if an error occurs. If the message is valid, this field is blank.

You can slide the columns around to rearrange their order. To move a column header, left-click and hold while you move the column to it's new location. You can also click any column to re-sort it, or use CTRL + left-click to sort on multiple columns.



If you right-click on a column header, a menu displays that allows you to automatically resize one or all columns (based on the displayed data in the columns), as well as to indicate which columns you would like to display.

Clicking on any column header causes the data to be sorted using that header. Click once to sort the column in ascending order. Click a again to sort the column in descending order. The third click clears the sort.

If you want to sort on multiple columns, use the CTRL key when clicking the column headers.

Receive Real-Time Data in the Multicast Transcript

If you select **Realtime Update**, the screen updates dynamically as messages are processed by the tool. However, real-time updating can cause application slowdown.

Filter Multicast Transcript Records Quickly

Just below the transcript options is a magnifying glass and entry field that allows you to filter messages based on entered data.

Clicking on the magnifying glass gives you a menu that you can use to provide additional selection criteria.

This powerful tool allows you to immediately view any set of messages that you can imagine, limited only by the data displayed in the columns.

View the Content of a Multicast Message

By double-clicking on any row, you can examine the details of the multicast message. The multicast message detail view lets you switch between a text and a hexadecimal view. You can also browse through the multicast transcript list while in detail view.

A "secret" search utility helps you find keywords within the message. Simply click in the window where the XML is displayed and start typing. The tool instantly jumps to the first match of the entered string. The up and down arrows will move you to the next or previous match of the entered string. This feature becomes very handy when you want to find data in large XML messages. The search utility works whether you are displaying hexadecimal or text.

To view the content of a multicast message:

1. Double-click the message you want to view.
   Several fields provide you with information about the selected message.
   - **Date Received -** Date and time message is received by the tool.
   - **Multicast Location -** Multicast address or other transport-specific parameters.
   - **Current Key** - Key used to authenticate message.
   - **UMAC -** Message Authentication Code using Universal hashing.
   - **UMAC Nonces -** The core security requirement of the UMAC is the notion of the nonce. MTP nonces are a combination of (Message ID + Frame Index + 0x0 + 0x0).
   - **Message ID -** Identification number of the message.
   - **Frame Index** - Index number of the frame.
   - **Total Frame Count -** Total number of frames in message.
   - **Frame Length** - Length of the frame contents.
   - **Comment - Indicates an invalid UMAC or an invalid frame length. If the message is valid, this field is blank**
2. On the Message tab, click **Text View** to view the message in either text format, or click Hex View to view the message in hexadecimal format.
3. Click **Ciphertext** to view the message payload. If the message is encrypted, the information displays in hexadecimal format.
4. Click **Previous** and **Next** to navigate through the Multicast transcript list while in message detail view.
5. Click **OK** to return to the multicast transcript.

Search the Content of a Multicast Message

The Search Content option lets you search for keywords within all messages currently displayed in the Multicast transcript.

1. Click **Search Content**.
2. Type the information you want to search on.

3. Click **Proceed**, or click **Cancel** to return to the Multicast Transcript Control. A pop-up window displays all messages containing the text you entered.
4. Click any message to display the Multicast message details along with the XML message text and frame payload.
5. Click **Back** to return to the Multicast Transcript Control.

Clear the Multicast Transcript Display

The Clear Display option lets you clear all messages from the Multicast transcript. This option does not remove messages from the Multicast transcript database.

1. Click **Clear Display** to remove all messages from the current view.

Clear the Multicast Transcript Database

The Clear DB option lets you remove all messages from the Multicast transcript database. Note that this action cannot be undone.

1. Click **Clear DB**.
2. Click **Yes** to remove all data from the transcript database, or click **No** to return to the transcript without clearing the database.

Clear the Multicast Listeners

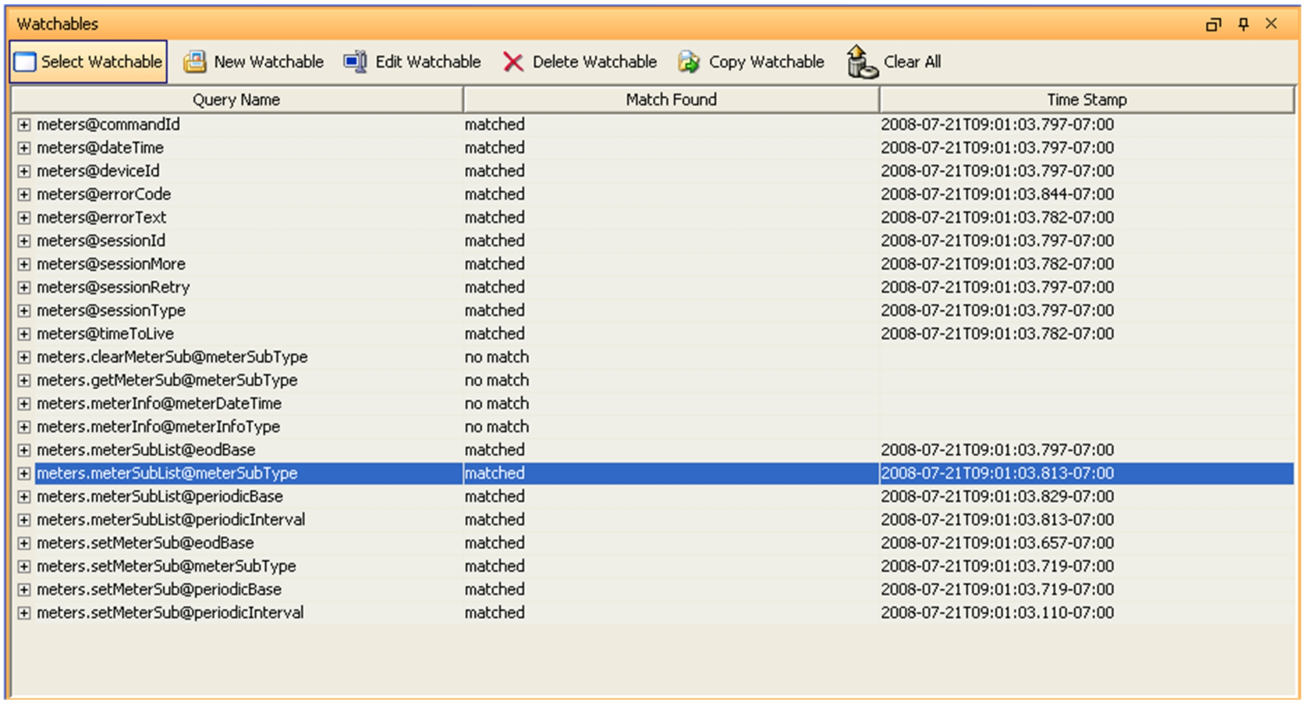
The Clear Multicast Listeners option lets you remove all endpoint multicast listeners from the Multicast transcript database. Note that this action cannot be undone.

1. Click **Clear Multicast Listeners**.
2. Click **Yes** to remove all endpoint multicast listeners, or click **No** to return to the transcript without removing the listeners.

About Watchables

The Watchables object lets you look for a specific attribute or even a specific value in that attribute. Simply specify a query to watch for particular attributes occurring in the XML stream that flows into and out of the application.

The Watchables feature is based on XML Path Language (XPath) queries. A Watchable will select all messages that meet the XPath criteria. With RadBlue Watchables, you can select from a list of pre-defined XPath criteria, modify the criteria of an existing XPath query, or create your own XPath query.



Query Name	Match Found	Time Stamp
⊞ meters@commandId	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@dateTime	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@deviceId	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@errorCode	matched	2008-07-21T09:01:03.844-07:00
⊞ meters@errorText	matched	2008-07-21T09:01:03.782-07:00
⊞ meters@sessionId	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@sessionMore	matched	2008-07-21T09:01:03.782-07:00
⊞ meters@sessionRetry	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@sessionType	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@timeToLive	matched	2008-07-21T09:01:03.782-07:00
⊞ meters.clearMeterSub@meterSubType	no match	
⊞ meters.getMeterSub@meterSubType	no match	
⊞ meters.meterInfo@meterDateTime	no match	
⊞ meters.meterInfo@meterInfoType	no match	
⊞ meters.meterSubList@eodBase	matched	2008-07-21T09:01:03.797-07:00
⊞ meters.meterSubList@meterSubType	matched	2008-07-21T09:01:03.813-07:00
⊞ meters.meterSubList@periodicBase	matched	2008-07-21T09:01:03.829-07:00
⊞ meters.meterSubList@periodicInterval	matched	2008-07-21T09:01:03.813-07:00
⊞ meters.setMeterSub@eodBase	matched	2008-07-21T09:01:03.657-07:00
⊞ meters.setMeterSub@meterSubType	matched	2008-07-21T09:01:03.719-07:00
⊞ meters.setMeterSub@periodicBase	matched	2008-07-21T09:01:03.719-07:00
⊞ meters.setMeterSub@periodicInterval	matched	2008-07-21T09:01:03.110-07:00

About XPath Expressions

The basic XPath expression for RadBlue tools contains a message and a body element. These elements must be defined in the expression. You can then use class, message, attribute name and attribute value elements to track specific messages and their content.

XPath Expression Format

/g2s:g2sMessage/g2s:g2sBody/g2s:cabinet/g2s:cabinetStatus/@g2s:denomId="526059076"

g2s:g2sMessage	g2s:g2sBody	g2s:cabinet	g2s:cabinetStatus	@g2s:denomId	="526059076"
message	body	class	message type	attribute name	attribute value

Sample XPath Expressions

Watchable	XPath Expression
EGM ID	/g2s:g2sMessage/g2s:g2sBody/@g2s:egmId
Specific EGM ID	/g2s:g2sMessage/g2s:g2sBody/@g2s:egmId=“RBG_12345”
Comms State	/g2s:g2sMessage/g2s:g2sBody/g2s:communications/g2s:commsStatus/@g2s:commsState
EGM Location	/g2s:g2sMessage/g2s:g2sBody/g2s:communications/g2s:commsOnLine/@g2s:egmLocation
Paytable ID	/g2s:g2sMessage/g2s:g2sBody/g2s:cabinet/g2s:cabinetStatus/@g2s:paytableId
Theme ID	/g2s:g2sMessage/g2s:g2sBody/g2s:cabinet/g2s:cabinetStatus/@g2s:themeId
Cabinet Status	/g2s:g2sMessage/g2s:g2sBody/g2s:eventHandler/g2s:eventReport/g2s:deviceList/g2s:statusInfo/g2s:cabinetStatus

XPath References

- World Wide Web Consortium (W3C), XPath 2.0 web site: http://www.w3.org/TR/xpath20/
- Kay, Michael. *XPath 2.0 Programmer’s Reference (Programmer to Programmer)*. Indianapolis, IN: Wiley Publishing, Inc., 2004.

Boolean Expression Usage

XPath allows boolean expressions. For example:

```
if ( count(/g2s:g2sMessage/g2s:g2sBody/g2s:eventHandler/g2s:eventReport/
g2s:meterList/g2s:meterInfo[2]/g2s:deviceMeters[1]/g2s:* ) = 1 ) then true() else false()
```

The above sample expression tracks messages that contain one, and only one, deviceMeter element under a second meterInfo element. Note that the asterisk (*) indicates that the value of deviceMeters can match anything in that element.

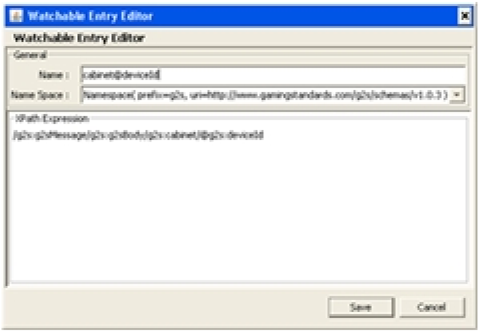
Clear All Watchable Data

Click Clear All to remove all of the matched data from the display.

Copy a Watchable

Copy Watchable allows you to make a copy of one of the queries, in case you want to modify your query just a bit more, but don't want to lose the prior one.

1. Select the watchable you want to copy from the list of active queries.
2. Click **Copy Watchable**.



Name : cabinet@deviceId

Name Space : Namespace(prefix=g2s, uri=http://www.gamingstandards.com/g2s/schemas/v1.0.3)

/g2s:g2sMessage/g2s:g2sBody/g2s:cabinet/@g2s:deviceId

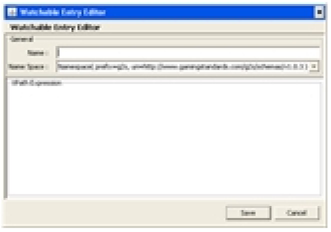
3. Click in the **Name** dialog box, and type a new name for the Watchable.
4. Click **Save**.

Create a New Watchable

Use New Watchable to write your own XPath expressions to further refine your search. For example, you can look for a particular value in a particular attribute (**../@g2s:deviceClass="G2S_all"**).

The XPath specification is available online through the World Wide Web Consortium (W3C). When creating a new Watchable, note that the Watchable namespace must correspond to the GSA schema that the Watchable applies to.

1. Click **New Watchable**.



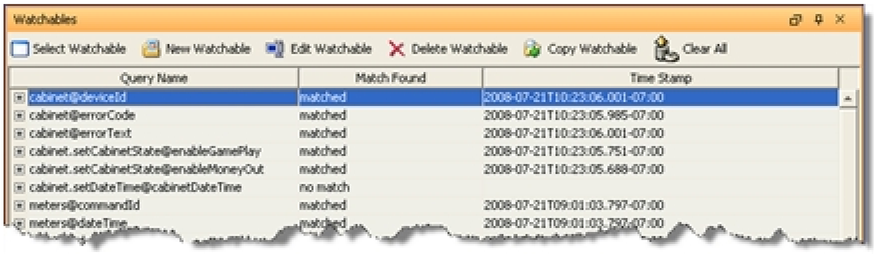
Name :

Name Space : Namespace[prefix=g2s, uri=http://www.gamingstandards.com/g2s/schemas/v1.0.3]

2. Type the **Name** of the new Watchable.
3. Click the drop-down arrow, and select a **Name Space** for the new Watchable. The namespace provides context for the query.
4. Type the XPath query in the **XPath Expression** text box.
5. Click **Save** to make the newly created Watchable available in the Watchable list.

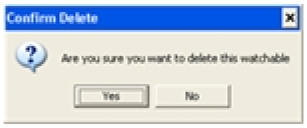
Delete a Watchable

1. Select the watchable you want to delete from the list of active queries.



Query Name	Match Found	Time Stamp
cabinet@deviceId	matched	2008-07-21T10:23:06.001-07:00
cabinet@errorCode	matched	2008-07-21T10:23:05.985-07:00
cabinet@errorText	matched	2008-07-21T10:23:06.001-07:00
cabinet.setCabinetState@enableGamePlay	matched	2008-07-21T10:23:05.751-07:00
cabinet.setCabinetState@enableMoneyOut	matched	2008-07-21T10:23:05.688-07:00
cabinet.setDateTime@cabinetDateTime	no match	
meters@commandId	matched	2008-07-21T09:01:03.797-07:00
meters@dateTime	matched	2008-07-21T09:01:03.797-07:00

2. Click **Delete Watchable**.



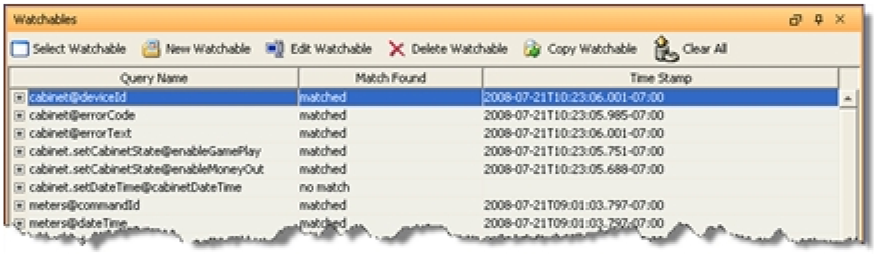
Are you sure you want to delete this watchable

3. Click **Yes** to delete the selected XPath query.

Edit a Watchable

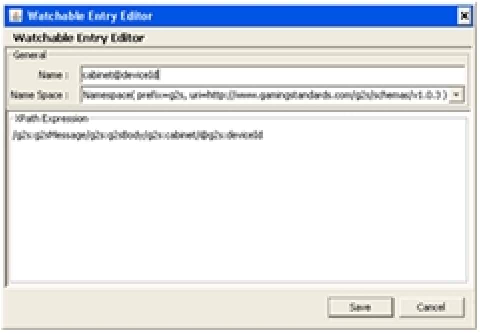
Edit Watchable allows you to modify the name or expression of the selected XPath query. Note that you must restart the tool before your changes will take effect.

1. Select the watchable you want to edit from the list of active queries.



Query Name	Match Found	Time Stamp
cabinet@deviceId	matched	2008-07-21T10:23:06.001-07:00
cabinet@errorCode	matched	2008-07-21T10:23:05.985-07:00
cabinet@errorText	matched	2008-07-21T10:23:06.001-07:00
cabinet.setCabinetState@enableGamePlay	matched	2008-07-21T10:23:05.751-07:00
cabinet.setCabinetState@enableMoneyOut	matched	2008-07-21T10:23:05.688-07:00
cabinet.setDateTime@cabinetDateTime	no match	
meters@commandId	matched	2008-07-21T09:01:03.797-07:00
meters@dateTime	matched	2008-07-21T09:01:03.797-07:00

2. Click **Edit Watchable**.



Name : cabinet@deviceId

Name Space : Namespace(prefix=g2s, uri=http://www.gamingstandards.com/g2s/schemas/v1.0.3)

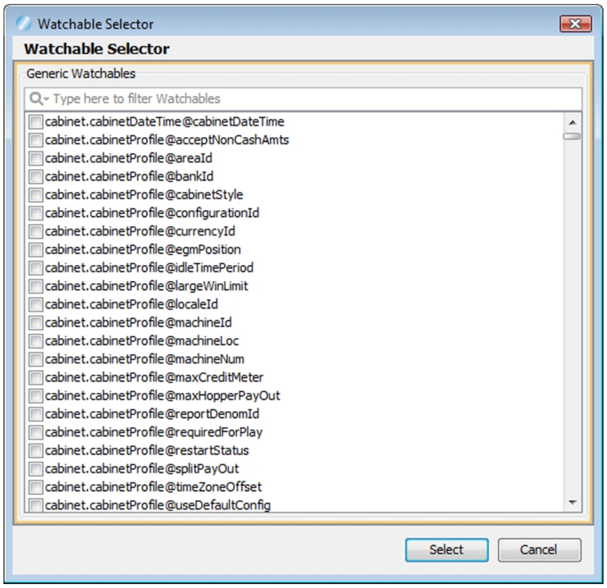
/g2s:g2sMessage/g2s:g2sBody/g2s:cabinet/@g2s:deviceId

3. Modify the selected Watchable as needed.
4. Click **Save**.

Select Attribute(s) to Track in Watchables

Select Watchable provides a searchable list of all standard attributes in G2S or S2S. Once you select the Watchable criteria, all messages from that point forward will be matched against the Watchable to see if it should be displayed.

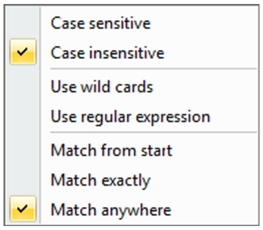
1. Click **Select Watchable**.



□ cabinet.cabinetDateTime@cabinetDateTime
□ cabinet.cabinetProfile@acceptNonCashAmts
□ cabinet.cabinetProfile@areaId
□ cabinet.cabinetProfile@bankId
□ cabinet.cabinetProfile@cabinetStyle
□ cabinet.cabinetProfile@configurationId
□ cabinet.cabinetProfile@currencyId
□ cabinet.cabinetProfile@egmPosition
□ cabinet.cabinetProfile@idleTimePeriod
□ cabinet.cabinetProfile@largeWinLimit
□ cabinet.cabinetProfile@localeId
□ cabinet.cabinetProfile@machineId
□ cabinet.cabinetProfile@machineLoc
□ cabinet.cabinetProfile@machineNum
□ cabinet.cabinetProfile@maxCreditMeter
□ cabinet.cabinetProfile@maxHopperPayOut
□ cabinet.cabinetProfile@reportDenomId
□ cabinet.cabinetProfile@requiredForPlay
□ cabinet.cabinetProfile@restartStatus
□ cabinet.cabinetProfile@splitPayOut
□ cabinet.cabinetProfile@timeZoneOffset
□ cabinet.cabinetProfile@useDefaultConfig

2. Select the attribute(s) you want to track.

   You can quickly filter the available attributes by clicking in the text box at the top of the screen and typing a keyword or characters. To configure additional selection criteria, click the magnifying glass to the left of the text box.



2. Click **Select** to add the selected attribute(s) to the list of attributes that the application is tracking.

   The tool creates a sample xPath query that can then be modified. See Create a New Watchable.

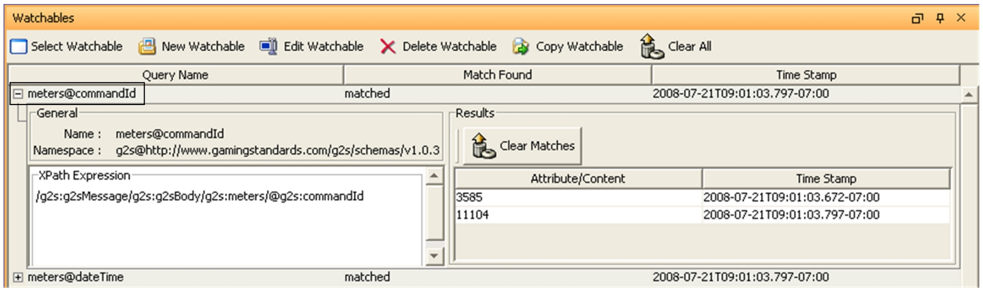
View a Watchable

For each query, the XPath expression, attribute, and date received are displayed. If no commands were received with the attribute specified in a specific query, “no match” is displayed.

For some of the more complex commands, such as the meterInfo command, the application provides tabs and tables for easy navigation.

First, select a meter group (GameDenom, Device, Currency, or Wager) and then select the individual device whose meters you want to examine from the list of those reported by the EGM. By clicking the “plus” (+) sign, you cause the list of the devices meters to be expanded as per the above example. For long meter lists, a scroll bar is provided on the right to easily move through the list.

1. Select the entry you want to view, and click the “plus” (+) sign in front of the entry.



Query Name	Match Found	Time Stamp
⊟ meters@commandId	matched	2008-07-21T09:01:03.797-07:00
⊞ meters@dateTime	matched	2008-07-21T09:01:03.797-07:00

General

Name : meters@commandId
Namespace : g2s@http://www.gamingstandards.com/g2s/schemas/v1.0.3

XPath Expression

/g2s:g2sMessage/g2s:g2sBody/g2s:meters/@g2s:commandId

Results

Attribute/Content	Time Stamp
3585	2008-07-21T09:01:03.672-07:00
11104	2008-07-21T09:01:03.797-07:00

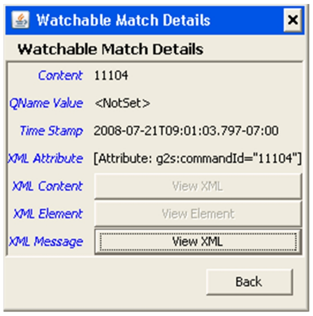
General

- **Name** - Watchable designation.
- **Namespace** - Namespace that was used in the XPath expression.
- **XPath Expression** - XPath query that the attribute or message content is matched to.

Results

- **Attribute/Content** - Attribute or message content of the matched XPath query.
- **Time Stamp** - Time and date the message was sent.

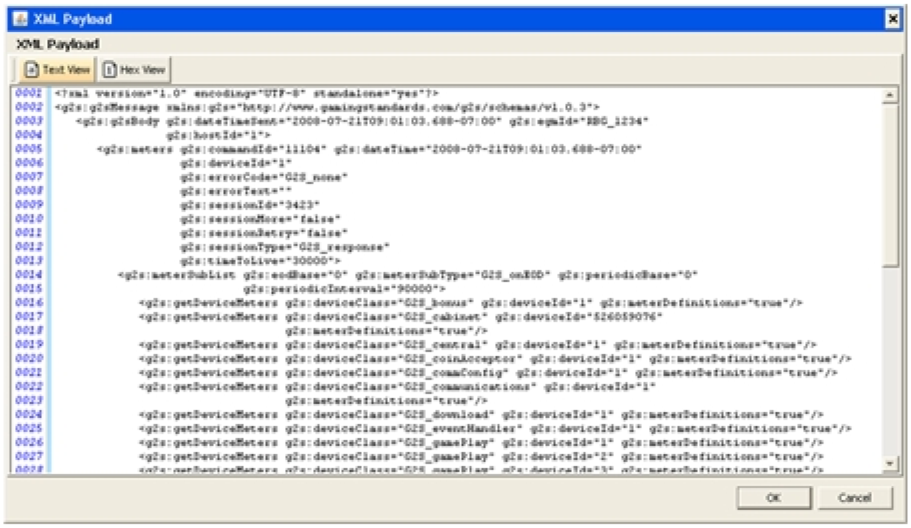
2. In the **Results** section, double-click the message you want to view.



Watchable Match Details

Content	11104
QName Value	<NotSet>
Time Stamp	2008-07-21T09:01:03.797-07:00
XML Attribute	[Attribute: g2s:commandId="11104"]
XML Content	View XML
XML Element	View Element
XML Message	View XML

3. To view the message's XML content, click **View XML**.



```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<g2s:g2sMessage xmlns:g2s="http://www.gamingstandards.com/g2s/schemas/v1.0.3">
   <g2s:g2sBody g2s:dateTimeSent="2008-07-21T09:01:03.688-07:00" g2s:egmId="RBG_1234"
                g2s:hostId="1">
      <g2s:meters g2s:commandId="11104" g2s:dateTime="2008-07-21T09:01:03.688-07:00"
                  g2s:deviceId="1"
                  g2s:errorCode="G2S_none"
                  g2s:errorText=""
                  g2s:sessionId="3423"
                  g2s:sessionMore="false"
                  g2s:sessionRetry="false"
                  g2s:sessionType="G2S_response"
                  g2s:timeToLive="30000">
         <g2s:meterSubList g2s:eodBase="0" g2s:meterSubType="G2S_onEOD" g2s:periodicBase="0"
                           g2s:periodicInterval="90000">
            <g2s:getDeviceMeters g2s:deviceClass="G2S_bonus" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_cabinet" g2s:deviceId="526059076"
                                 g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_central" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_coinAcceptor" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_commConfig" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_communications" g2s:deviceId="1"
                                 g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_download" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_eventHandler" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_gamePlay" g2s:deviceId="1" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_gamePlay" g2s:deviceId="2" g2s:meterDefinitions="true"/>
            <g2s:getDeviceMeters g2s:deviceClass="G2S_gamePlay" g2s:deviceId="3" g2s:meterDefinitions="true"/>
```

4. To view the message content in hexadecimal format, click **Hex View**.

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.

You can specify the maximum number of lines included in the log through the Configure option under Tools on the menu bar.

1. Go to: **Tools > Configure** > **Desktop Options** > **Max Logger Lines**
2. Click **Desktop Options**.
3. In the **Max Logger Lines** field, type or select the maximum number of lines in the Debug Console.
4. Click **OK**.

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:

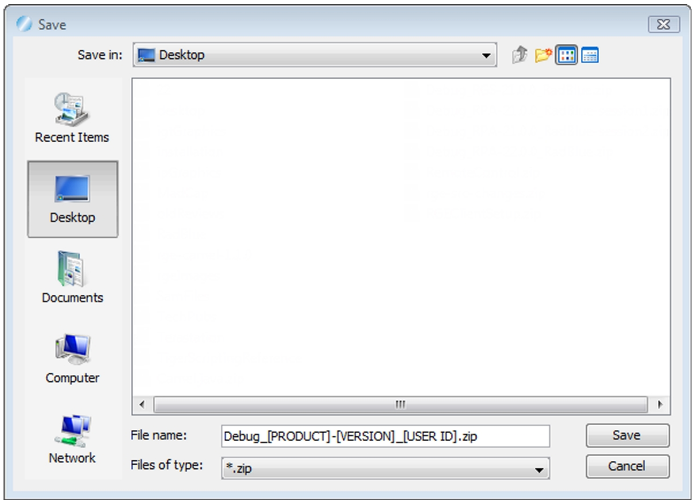
1. From the Debug Console, click **Filter**. The Logging Filter screen appears.
2. Click **Select All** to select all message types.
   or

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**.



Save in: Desktop

File name: Debug_[PRODUCT]-[VERSION]_[USER ID].zip

Files of type: *.zip

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.

Configuring Desktop Options

Desktop Options define default application views, which are comprised of one or more available controls. This screen also allows you to define the amount of data displayed in specific transcripts and views.

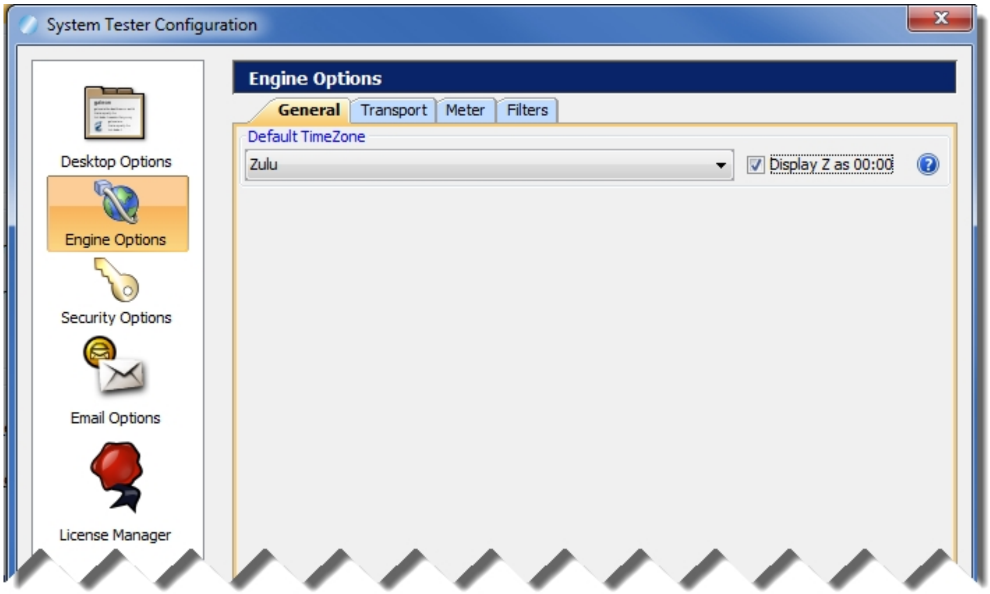
- SmartEgm Config
- **Transcript and Log Messages Displayed**

  This section allows you to specify the amount of data used for various transcripts and views. Increasing the data sizes increases the memory used by the tool.

  - **Max Transcript Messages -** Maximum number of G2S transcript messages displayed.
  - **Max Soap Transcript Messages -** Maximum number of SOAP transcript messages displayed.
  - **Max Logger Lines -** Maximum number of lines in the debug log stored in the database.
  - **Max Watcher Data Versions -** Maximum number of matches for each Watchable stored in the database.

Configuring RST Engine Options

Engine Options allow you to automate some functions of the RST engine. Engine options are grouped functionally: General, Transport, Meter, and Filters. Click any tab to view the options for that group.



RST Engine Options screen.

General

From the General tab on the Engine Options configuration screen, you can change the timezone used in RST through the *Default TimeZone* configuration option (**Tools > Configure > Engine Options > General**). You can also set a specific time offset from the system clock that is applied to the dateTime stamp on all outgoing messages.

Default Time Zone - Select the appropriate time zone for your area by country/city or by Coordinated Universal Time (UTC)/ Zulu/ Greenwich Mean Time (GMT).

- You can select the timezones in the following ways:
  - **Scroll** through the list and select your country/city, or timezone preference — UTC / Zulu / GMT.
  - **Type** in the country/city or timezone.
  - **Type a portion** of the country and select the rest, for example, If you want the time for *Los Angeles, California*, you can type: **America/L** and this option finds Los Angeles.

- Select the *Display Z* checkbox if you *do not* want to see a "Z" at the end of the date/time stamp in the Transcript when selecting UTC / Zulu / GMT. Selecting a new time zone requires a restart.

Current RST Offset [HH:MM:SS.MMM] - Set the RST time offset from the system clock, which is part of the *dateTime* calculation applied to all messages sent by RST. The calculation for the *dateTime* stamp is:

PC time + **configured offset** [Current RST Offset value] + **setDateTime offset**

Use this option if you want to test an incorrect *dateTime* value.

Filters

From the Filters tab on the Engine Options configuration screen, you can define settings for filtering data collected by RST. Once selected, the filter persists until it is cleared.

Note that you can filter all of the following commands through the Filters option on the Message Transcript. That filter, however, does not persist between runs of the tool.

- **Filter G2S ACKs from Transcript** - Select to exclude G2S `g2sAck` commands from the Message Transcript display.
- **Filter G2S Keep Alives from Transcript** - Select to exclude G2S `communications.keepAlive` commands from the Message Transcript display.
- **Filter G2S Set Progressive Values from Transcript** - Select to exclude G2S `progressive.setProgressiveValue` commands from the Message Transcript display.
- **Filter Multicast G2S Bonus Activities from Transcript** - Select to exclude G2S `bonus.bonusActivity` commands from the Message Transcript display.

Meter

From the Meter tab on the Engine Options configuration screen, you can define RST settings related to meters.

- **Meter Rollover Value** - Type or select the reset value for all Amount (Amt) and Count (Cnt) meters. The default is **999,999,999,999,999**.

Transport

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

- **Bind To** - Click the drop-down arrow, and select the IP Address that you want RST to use for communications. If the RST, EGM and host are all running on the same computer, select **127.0.0.1** (localhost).

- **SOAP Port** - Enter the port that you want RST to use for communications. We recommend that you do not change the SOAP port unless you have a port conflict.
- **SSL SOAP Port** - Enter the port that you want RST to use for SSL-enabled communications.

  Note: RST must be restarted when you make a change to the SOAP port number.
- **Protocol** - Click the drop-down arrow, and select either the HTTP protocol or HTTPS protocol.
- **My URL** - Type the URL for the current RST instance.
- **Remote Control** - URL of the RST REST interface (Remote Control). The REST interface is an optional module for RST. This field is *read-only*. To copy the URL, highlight the address and type **CTRL+C**.
- **Scratch Pad** - URL of the example RST REST interface. The REST interface is an optional module for RST. This field is *read-only*. To copy the URL, highlight the address and type **CTRL+C**.
- **Enable Get Transport Options** - If this option is selected, RST requests `getTransportOptions` before the `communications.commsOnline` command. If this option is cleared, RST assumes **No GZIP** support. By default, this option is selected.
- **Enable GZIP** - If selected, RST supports GZIP compression when exchanging messages, in either direction, between EGM and host.
- **SOAP Timeout** - Set the SOAP connection timeout value. The default is **30 seconds**.

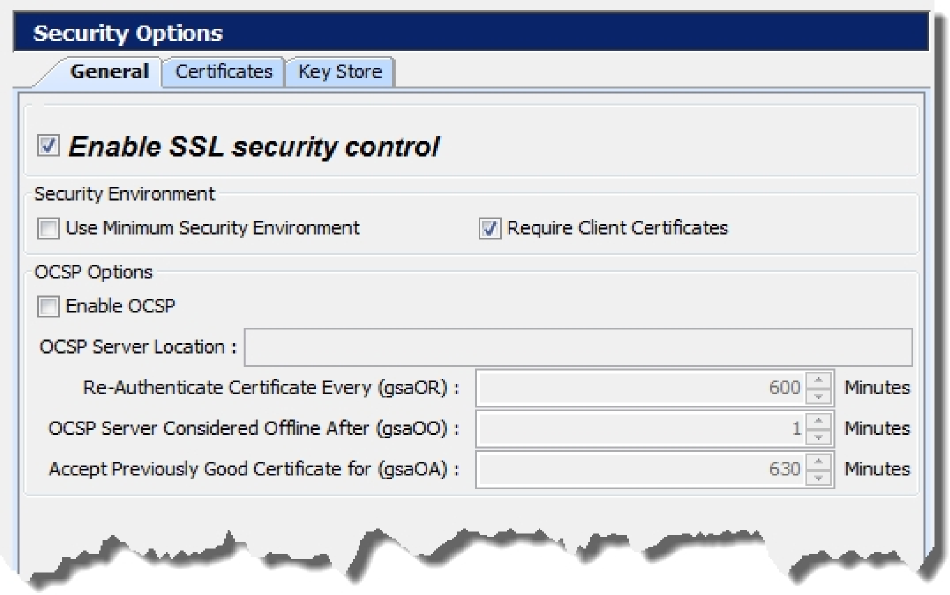
- **Enable Get Transport Options -** Select to execute the `getTransportOptions` WSDL call automatically at start-up.
- **SOAP Timeout -** Enter the time using **mm:ss:sss** format or keep the 30-second default.

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.



☑ Enable SSL security control

Security Environment

□ Use Minimum Security Environment ☑ Require Client Certificates

OCSP Options

□ Enable OCSP

OCSP Server Location :

Re-Authenticate Certificate Every (gsaOR) : 600 Minutes

OCSP Server Considered Offline After (gsaOO) : 1 Minutes

Accept Previously Good Certificate for (gsaOA) : 630 Minutes

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:

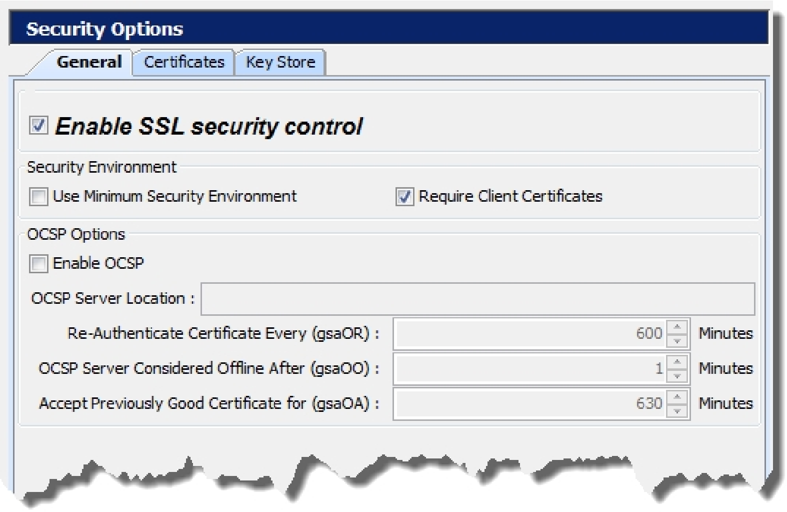
- *Signed by trusted certificate authority?*
- *Is current time/date within the period of validity (effective and expired date)?*
- *Is issuer signature correct?*

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**.



☑ *Enable SSL security control*

Security Environment

☐ Use Minimum Security Environment ☑ Require Client Certificates

OCSP Options

☐ Enable OCSP

OCSP Server Location :

Re-Authenticate Certificate Every (gsaOR) : 600 Minutes

OCSP Server Considered Offline After (gsaOO) : 1 Minutes

Accept Previously Good Certificate for (gsaOA) : 630 Minutes

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.

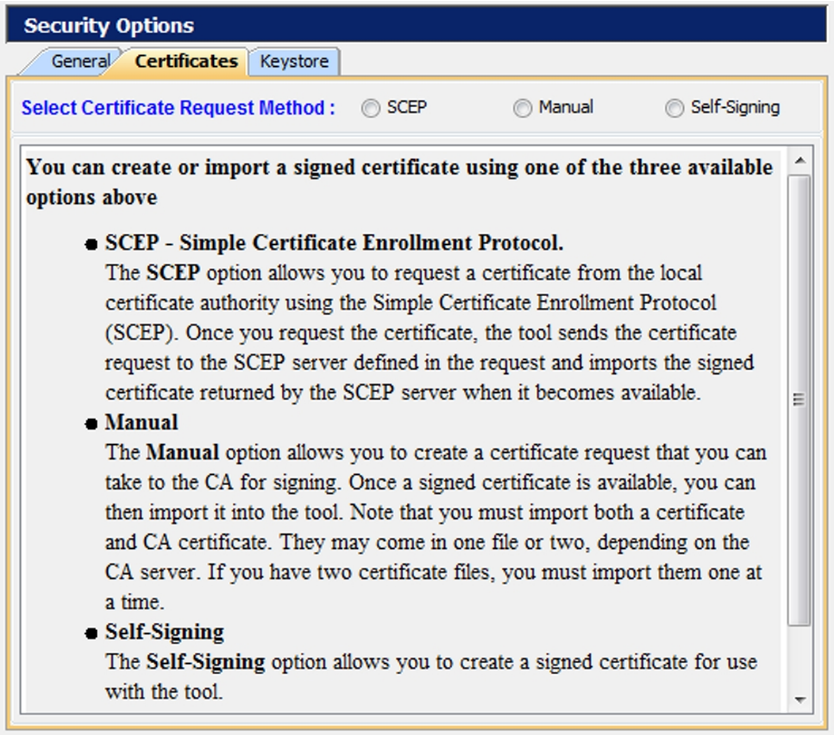
When you enable OCSP, you must configure the following options:

- **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



Select Certificate Request Method : ○ SCEP ○ Manual ○ Self-Signing

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

- **SCEP - Simple Certificate Enrollment Protocol.**
  The **SCEP** option allows you to 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.
- **Manual**
  The **Manual** option allows you to create a certificate request that you can take to the 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.
- **Self-Signing**
  The **Self-Signing** option allows you to create a signed certificate for use with the tool.

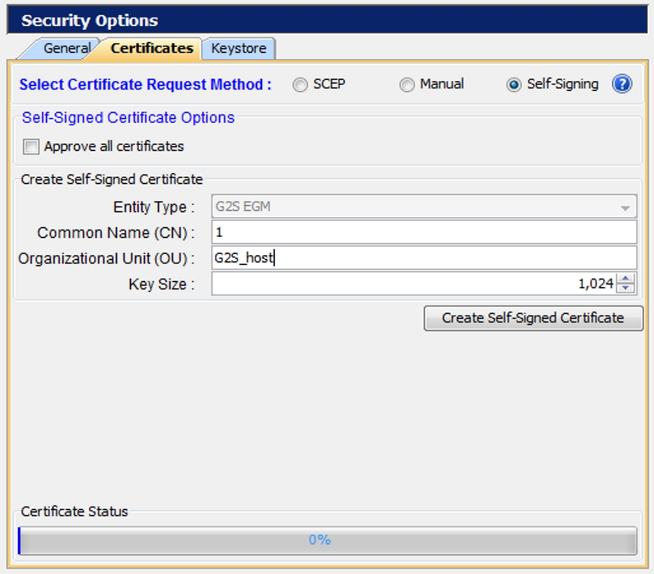
To access the certificate options:

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.

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**.


Security Options
General
Certificates
Keystore
Self-Signed Certificate Options
Create Self-Signed Certificate
Create Self-Signed Certificate
Certificate Status

Select Certificate Request Method : ○ SCEP ○ Manual ● Self-Signing

□ Approve all certificates

Entity Type : G2S EGM

Common Name (CN) : 1

Organizational Unit (OU) : G2S_host

Key Size : 1,024

0%

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.

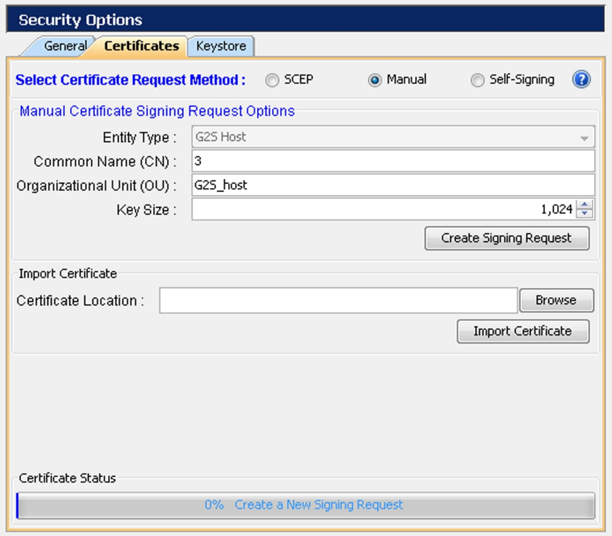
- **Organizational Unit** - Type the organizational unit (role) of the tool: G2S_host, G2S_egmProxy, G2S_egm, or Other G2S. By default, this field is populated with a value that corresponds to the Entity Type.
- **Key Size -** Click the drop-down arrow, and select the size of the key pair supported by 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.

Load a Manually Signed 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**.



Select Certificate Request Method : ○ SCEP ● Manual ○ Self-Signing

Entity Type : G2S Host

Common Name (CN) : 3

Organizational Unit (OU) : G2S_host

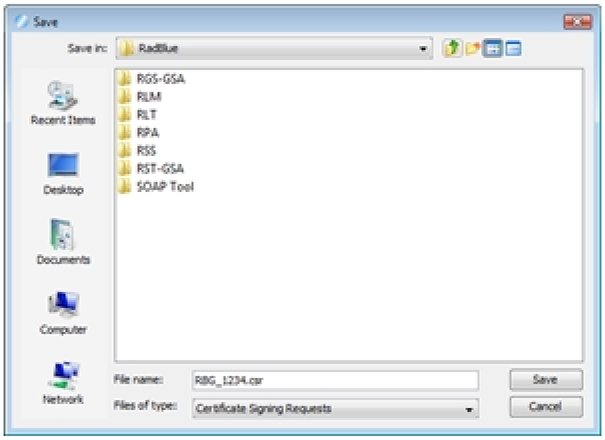
Key Size : 1,024

Certificate Location :

0% Create a New Signing Request

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.

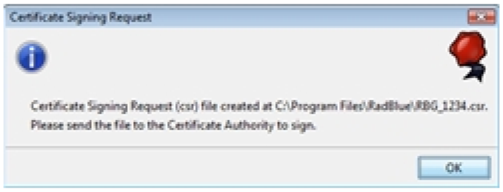
2. Click **Create Signing Request** to generate a signing request.



File name: RBG_1234.csr

Files of type: Certificate Signing Requests

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**.



Certificate Signing Request (csr) file created at C:\Program Files\RadBlue\RBG_1234.csr.
Please send the file to the Certificate Authority to sign.

6. Click **OK**.

   Notice that the Certificate Status is updated.

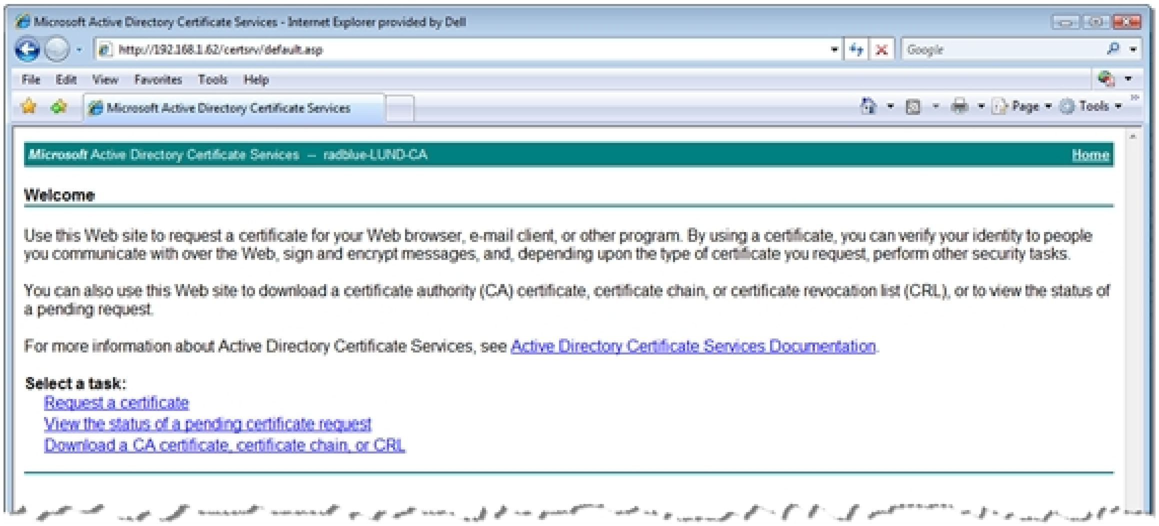
7. Depending on the certificate authority you are using, you must now use the certificate request you created to obtain a signed certificate from a certificate authority. For an example of how to obtain a signed certificate from Microsoft Active Directory Certificate Services, see Obtaining a Signed 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**.


Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/default.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page
Tools
Microsoft Active Directory Certificate Services -- radblue-LUND-CA
Home

Welcome

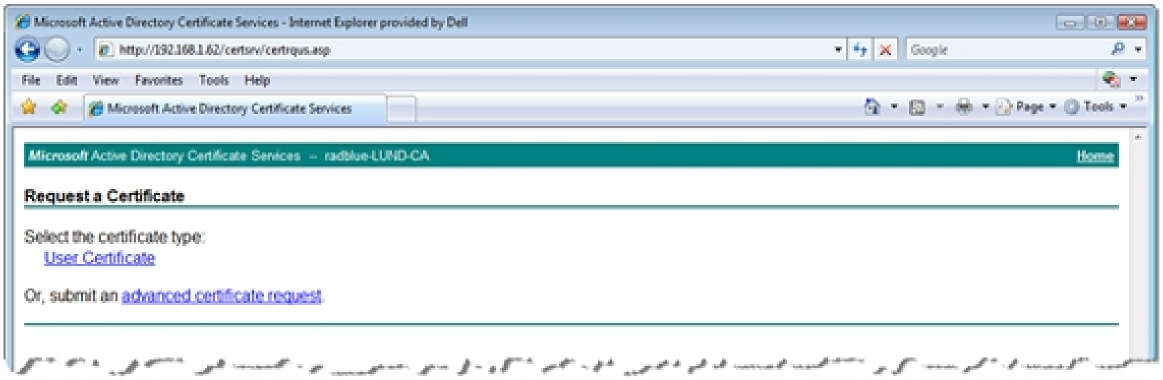
Use this Web site to request a certificate for your Web browser, e-mail client, or other program. By using a certificate, you can verify your identity to people you communicate with over the Web, sign and encrypt messages, and, depending upon the type of certificate you request, perform other security tasks.

You can also use this Web site to download a certificate authority (CA) certificate, certificate chain, or certificate revocation list (CRL), or to view the status of a pending request.

For more information about Active Directory Certificate Services, see Active Directory Certificate Services Documentation.

Select a task:
Request a certificate
View the status of a pending certificate request
Download a CA certificate, certificate chain, or CRL

4. Click **Request a certificate**.

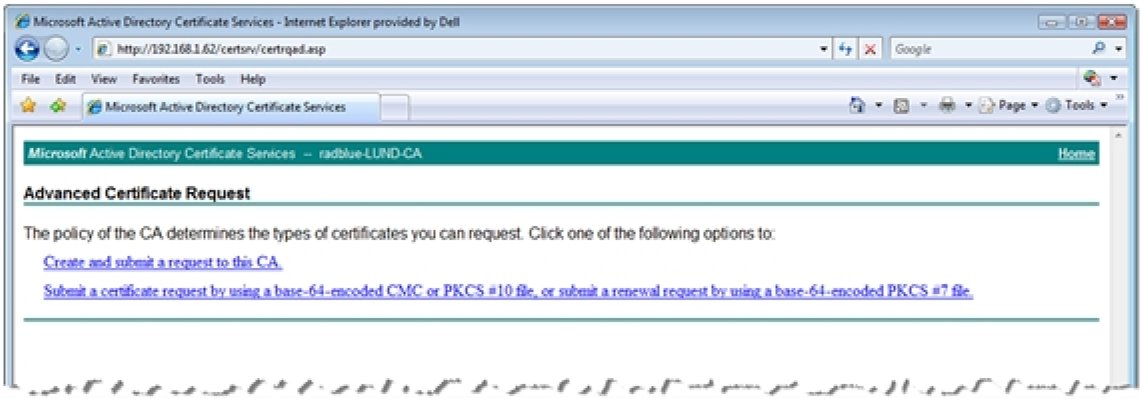

Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certrqus.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page
Tools
Microsoft Active Directory Certificate Services -- radblue-LUND-CA
Home

Request a Certificate

Select the certificate type:
User Certificate

Or, submit an advanced certificate request.

5. Click **advanced certificate request**.


Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certrqad.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page Tools
Microsoft Active Directory Certificate Services -- radblue-LUND-CA
Home

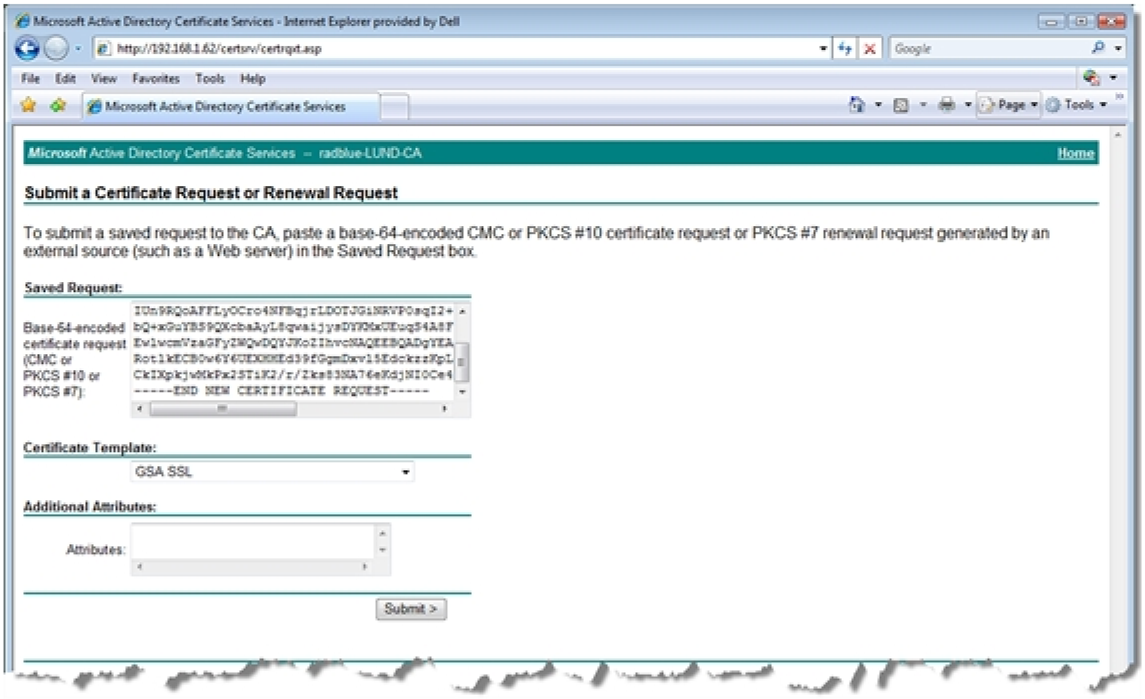
Advanced Certificate Request

The policy of the CA determines the types of certificates you can request. Click one of the following options to:

Create and submit a request to this CA.

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

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**.


Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certrqxt.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page Tools
Microsoft Active Directory Certificate Services -- radblue-LUND-CA
Home

Submit a Certificate Request or Renewal Request

To submit a saved request to the CA, paste a base-64-encoded CMC or PKCS #10 certificate request or PKCS #7 renewal request generated by an external source (such as a Web server) in the Saved Request box.

Saved Request:

Base-64-encoded certificate request (CMC or PKCS #10 or PKCS #7):

```
IUn9RQoAFFLyOCro4NFBqjrLDOTJGiNRVP0sqI2+
bQ+xGuYB59QXcbaAyL8qwaijysDYWMxUEuqS4A8F
EwlwcmVzaGFyZWQwDQYJKoZIhvcNAQEEBQADgYEA
Rot1kECB0w6Y6UEXHHEd39fGgmDav15EdckzzKpL
CkIXpkjwNkPx2STiK2/r/Zks83NA76eKdjNI0Ce4
-----END NEW CERTIFICATE REQUEST-----
```

Certificate Template:

GSA SSL

Additional Attributes:

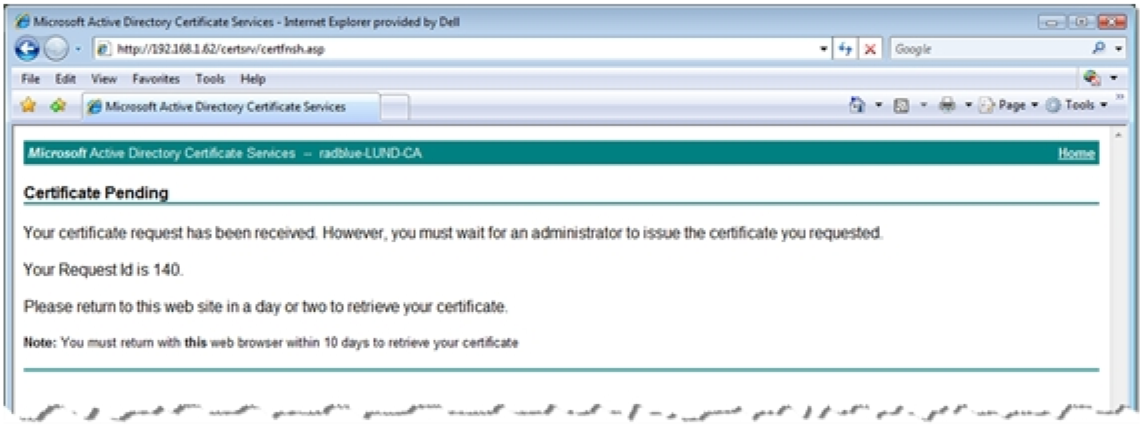
Attributes:

Submit >

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**.


Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certfnsh.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page Tools
Microsoft Active Directory Certificate Services -- radblue-LUND-CA
Home

Certificate Pending

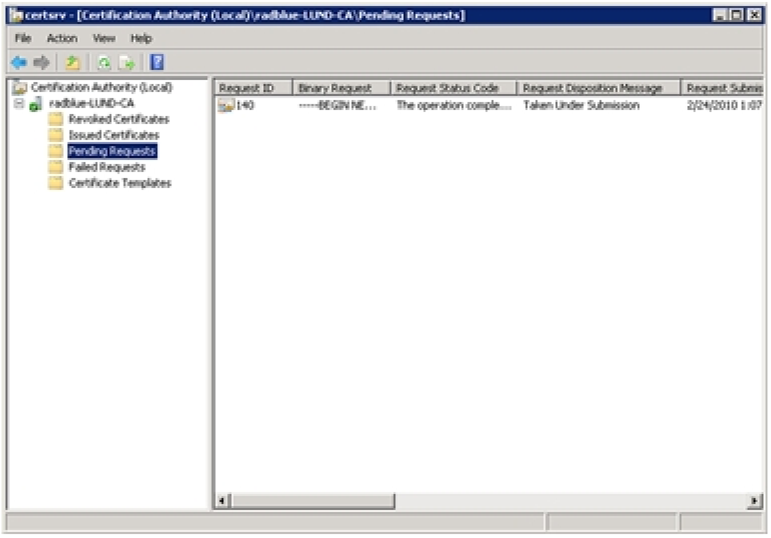
Your certificate request has been received. However, you must wait for an administrator to issue the certificate you requested.

Your Request Id is 140.

Please return to this web site in a day or two to retrieve your certificate.

Note: You must return with **this** web browser within 10 days to retrieve your certificate

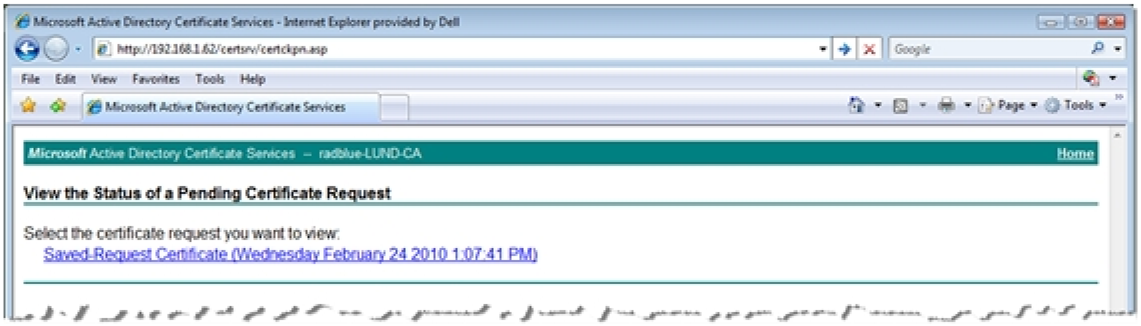
10. Note the **Request ID**. In this case, the Request ID is **140**.
11. Minimize, but do not close, the browser.
12. From the computer where Microsoft Active Directory Certificate Services is installed, go to: **Start** > **Administrative Tools** > **Certification Authority**



Request ID	Binary Request	Request Status Code	Request Disposition Message	Request Submis
140	-----BEGIN NE...	The operation comple...	Taken Under Submission	2/24/2010 1:07

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**.

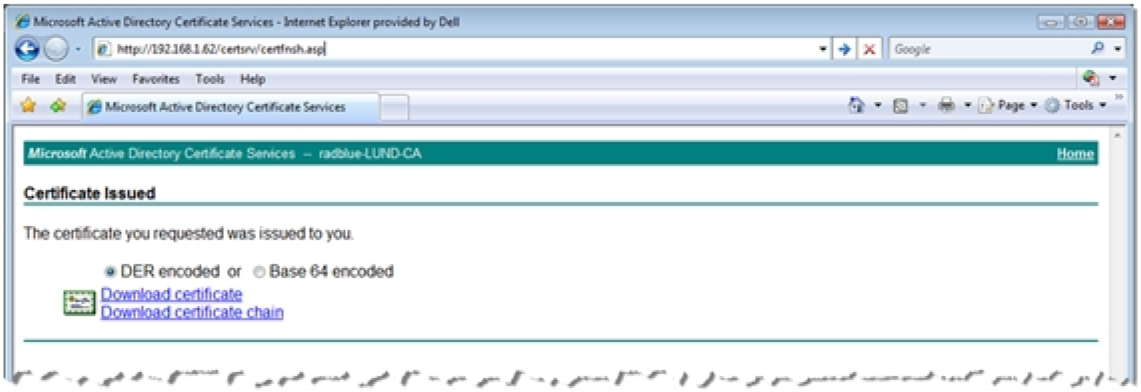

Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certckpn.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page Tools

Microsoft Active Directory Certificate Services -- radblue-LUND-CA Home

View the Status of a Pending Certificate Request

Select the certificate request you want to view:
Saved-Request Certificate (Wednesday February 24 2010 1:07:41 PM)

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


Microsoft Active Directory Certificate Services - Internet Explorer provided by Dell
http://192.168.1.62/certsrv/certfnsh.asp
Google
File Edit View Favorites Tools Help
Microsoft Active Directory Certificate Services
Page Tools

Microsoft Active Directory Certificate Services -- radblue-LUND-CA Home

Certificate Issued

The certificate you requested was issued to you.

● DER encoded or ○ Base 64 encoded
Download certificate
Download certificate chain

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.

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.)

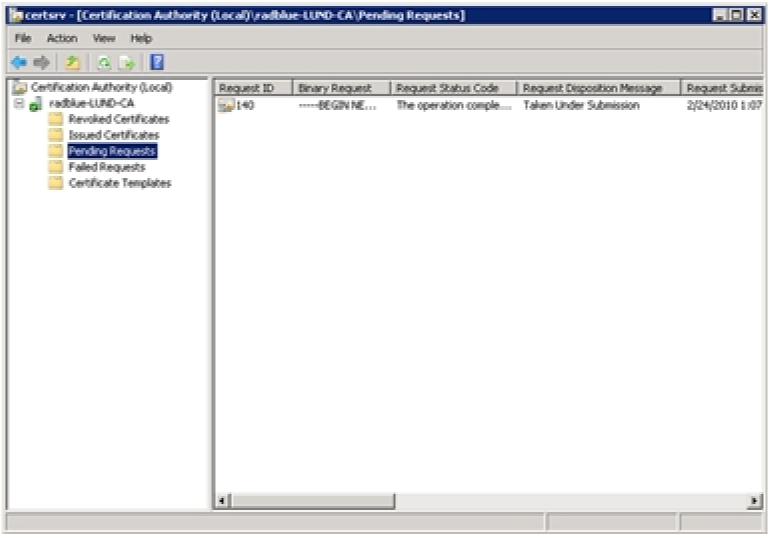
- **SCEP Server Polling Interval -** Type or select the interval, in milliseconds, in which the tool polls the certificate server until the tool's certificate request is approved.
- **Request SCEP Server Capabilities -** Select to request the options supported by the certificate authority server.
- **Request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



Request ID	Binary Request	Request Status Code	Request Disposition Message	Request Submis
140	-----BEGIN NE...	The operation comple...	Taken Under Submission	2/24/2010 1:07

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.

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**.
2. Select **Configuration**.
3. Click **Security Options** to display the Security Options screen.
4. Click the **Key Store** tab.
5. 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**.
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**. Scroll through the content, and click **OK** to close the certificate details window.

Configuring Email Options

With Email Options, you can configure outgoing mail server information so RST can automatically send an e-mail when a Tiger script has concluded.

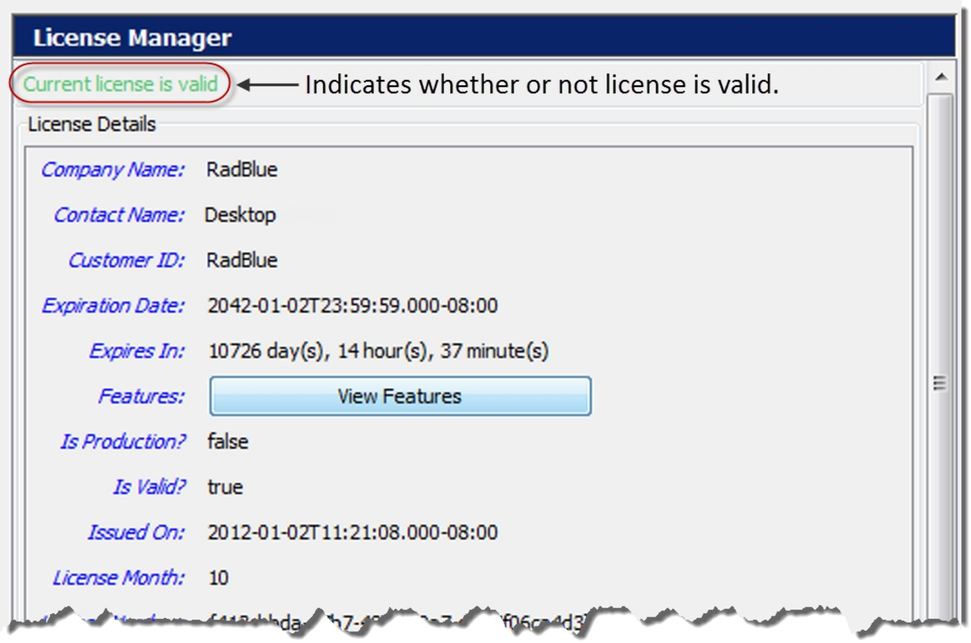
Note that the Tiger script must include one or more **toEmailAddressN** parameters, which defines the e-mail recipient(s). In addition, you can define when e-mails are generated by the tool by setting the Email Status option.

General Options

- **Email Status -** Select when e-mail notifications should be sent: Never, on the conclusion of every test regardless of outcome (Always), or only if the script fails (Failure).
- **SMTP Host Name -** Specify your outgoing mail host.
- **SMTP Host Port Number -** Specify the port number for outgoing e-mail.
- **SMTP Username (optional) -** Enter your e-mail user name.
- **SMTP Password (optional) -** Enter your e-mail user password.
- **From Address -** Specify the e-mail address of the contact person for the test.

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.



Current license is valid

License Details

Company Name: RadBlue

Contact Name: Desktop

Customer ID: RadBlue

Expiration Date: 2042-01-02T23:59:59.000-08:00

Expires In: 10726 day(s), 14 hour(s), 37 minute(s)

Features:

Is Production? false

Is Valid? true

Issued On: 2012-01-02T11:21:08.000-08:00

License Month: 10

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.

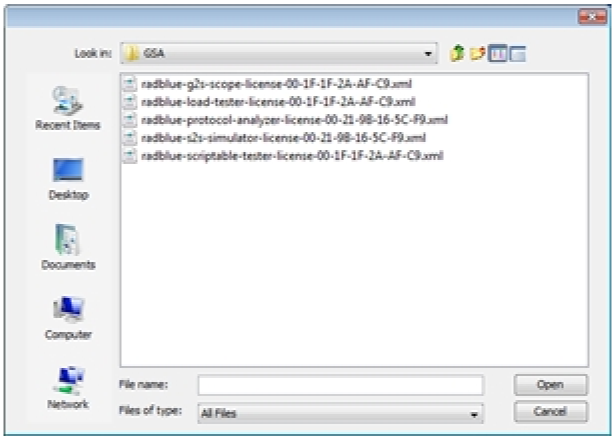
- **Location ID** - Location of purchasing organization.
- **MAC Address** - Physical address of computer on which the tool is installed.
- **Product Line Key** - Unique identifier of installed tool.
- **Product Name** - Name of licensed RadBlue product.

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.



Look in: GSA

radblue-g2s-scope-license-00-1F-1F-2A-AF-C9.xml
radblue-load-tester-license-00-1F-1F-2A-AF-C9.xml
radblue-protocol-analyzer-license-00-21-9B-16-5C-F9.xml
radblue-s2s-simulator-license-00-21-9B-16-5C-F9.xml
radblue-scriptable-tester-license-00-1F-1F-2A-AF-C9.xml

File name:
Files of type: All Files

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.

Index