



RadBlue G2S Scope Version 6

[Released: 09 DEC 2009]

Questions?

Log on to the RadBlue Forum: <http://radblue.mywowbb.com/>

Release Notes

Release Summary

In this release, we added a new *optional* Custom Configuration module. In addition to many general improvements and corrections, we've added new functionality to the voucher database, added multi-EGM support for progressives and made modifications to the joinMcast command on the Send Command screen.

New Features

- An optional Custom Configuration module is now available in RGS. The Custom Configuration module consists of the Response Manager, Script Runner and Startup Algorithm. Each one of these features provides you with additional testing flexibility.

Response Manager - The Response Manager lets you customize how RGS responds to G2S commands sent from the EGM.

Script Runner - Script Runner lets you run pre-configured, multi-command host operations (for example, the downloading and installation of a package). You can view the status of a script as it runs, and stop the script at any time.

Startup Algorithm - Startup Algorithms let you define the commands sent to the EGM when communications are initiated. You can choose to let the algorithm run to completion or select the step-through option to manipulate the startup timing, one command at a time.

Improvements

- The Compare option has been enhanced to highlight differences in the two messages being compared.
- The Transcript has been modified for improved record processing performance.
- You can now specify the starting pattern for validation IDs that are generated by RGS through the **Validation ID Pattern** option (**Configure > Engine Options**). The value entered (up to eight digits) will be used for the starting digits in the validation IDs generated by RGS..
- Three new buttons were added to the Voucher Database Control:
 - ♦ **Delete All Vouchers** deletes all vouchers from the voucher database.
 - ♦ **Delete All Redeemed Vouchers** deletes all redeemed vouchers from the database.
 - ♦ **Unassign All Redeemed Vouchers** allows redeemed vouchers to be reused.
- RGS now supports the G2S_APX013 (Command Not in commandId Order) error.

- On the **Send Command communications.joinMcast to G2S_communications** screen, port numbers are allocated automatically when the **Auto Populate** button is clicked. Select the multicast device, and then click **Auto Populate**. To ensure that the multicast group has a unique URL, select the multicast group and select **Auto Populate**.
- You can now send a `joinMcast` command from the **Send Command communications.joinMcast to G2S_communications** screen with no security parameters. To do this, you must clear the **Automate Multicast Host Behavior** option under the **Configure > Engine Options**.
- When sending a `communications.mCastJoin` command, the following rules are now enforced:
 - i. The pair of [Multicast ID, Multicast Location] has to be unique in its usage. That is, once the two values are used, they cannot be used in any other pair, either separate or apart. The two are now joined until the group is emptied of members.
 - ii. The combination of [Multicast ID, Multicast Location, EGM ID, Device Class, Device ID] must be unique for each join. This means that you cannot make the same device join the same group twice.
- An **Add** button has been added to the **Send Command printer.printTicket to G2S_printer** screen that lets you create print regions for the voucher being printed. The interface allows a minimum value of zero (0) to be entered in the Print Region and Template Index fields. No print region is sent is the Print Region value is **-1**.
- Progressives now support multiple EGMs. You can now have up to five EGMs contributing to a progressive.
- Port bind errors are now reported in the debug log.
- A **Force MSX003** button has been added to the Send Command object. Click this button to mark the communications channel as “lost”
- An **HTTP Receive Timeout** option has been added to the Engine Options screen. This option defines the maximum time, in milliseconds, for `g2sAck` commands to be received.
- On WAT transfers, the RGS now verifies that the requested amount is less than the maximum amount field. If the requested amount is greater than the maximum amount, RGS rejects the request.
- If the EGM sends a command for a device that was not in the `descriptorList` command, the RGS now returns a G2S_APX003 error (Invalid Device Identifier) to the EGM.
- Additional checking for GSA protocol required point-to-point class-level errors has been added. These errors are rejected by RGS and logged.
- The following options (**Configure > Engine Options**) have been added:
 - ◆ **Automatic Bonus Updates Using Multicast** - Select to deliver automatic bonus updates using multicast.
 - ◆ **The bonusActive flag** - Select to send the `bonusActive` attribute in the `bonusActive` command.
 - ◆ **Automatic Bonus Updates Interval** - Type, in milliseconds, the interval between sending automatic bonus updates. Zero (0) disables this option.

Corrections

- An issue with SSL connectivity has been corrected.
- The SSL port was not being written to the G2SConfig.xml file, so it could not be changed from the **Configure > Engine Options** screen. This issue has been corrected.
- An issue was corrected in which an error was caused by an `optionList` command that contained a set of empty option items for an option group.
- RGS was not properly handling EGM-initiated `optionList` requests. This issue has been corrected.
- The `joinMcast`, `mcastKeyUpdate` and `leaveMcast` commands no longer show the device class and device ID when displaying multicast group IDs in the send panel.
- Multiple multi-EGM errors have been corrected.
- RGS now resets the EGM device structure on a new `commsOnline` command.
- Previously, updated licenses could not be loaded through the user interface. When the license check is invalid (for example, due to expiration or tampering), RGS now loads the user interface without starting the engine, and the **Start Engine** option is disabled. You can now load a new valid license through the user interface.
- Multicast groups are now assigned port numbers by class so they do not have the same URL.
- The Player ID database (`id-database.xml`) is now updated when the `player.playerSessionEnd` command is received.
- In previous releases, an empty strings SOAP payload were not handled properly in the Transcript and transport. We have corrected this issue and empty strings now appear in the Transcript.



RadBlue G2S Scope Version 5.0

[Released: 04 NOV 2009]

Questions?

Log on to the RadBlue Forum: <http://radblue.mywowbb.com/>

Release Notes

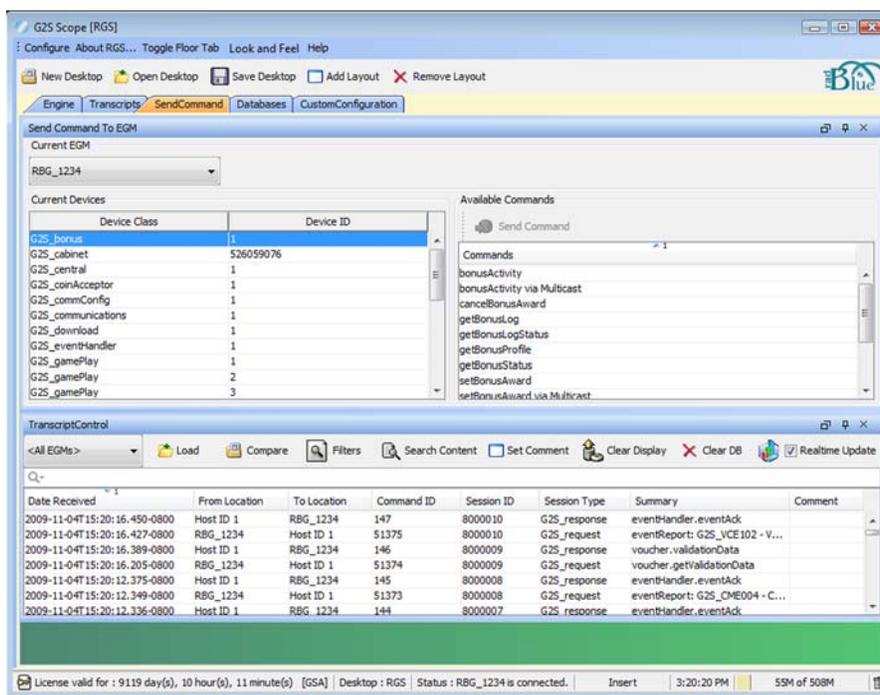
Release Summary

This month RGS - *Next Gen* officially becomes our main-line RGS offering. From this release forward, it will be versioned the same as other RadBlue products (5.0, 6.0, etc.). We've included a special [What's New](#) section to help you get up-to-speed on the changes as well as an [RGS 1.x vs. RGS 5.0](#) section to help you understand the technical differences between the new version of RGS and the old 1.x version.

Finally, for those of you who have been following the developments of the new RGS version, we've included the changes specific to this release.

Important Note!

We have also modified the RGS installer to determine the product version through your license. This allows us to have a *single installer* for multiple versions of the product. If you were previously using a custom installer, you can now use the general installer available on the RadBlue web site.



Not sure where to start? The [RGS, RST and RPA Quick Start](#) is a self-guided tutorial to get you up and running quickly on those products.

For detailed information on using the new version of RGS, see the [RGS User Guide](#).

What's New

While the new and improved RGS version contains all of the same features as the old version, we have made a few improvements that you should know about:

- **Multi-EGM support**
Compare any two messages from up to five EGMs, or a G2S EGM vs. a G2S SMIB, etc.
- **The Event Summary panel and Command Objects are gone - *Long live the Transcript!***
The new Transcript provides the same functionality in one location:
 - ◆ you can now use the Quick Filter to filter eventReport
 - ◆ the Summary column now shows the *eventId* and *eventText*
 - ◆ eventReport command objects are available on the message details screen (accessed when you double click on a row)
 - ◆ the message details screen now let's you toggle between an XML view of a command and a graphical command view (which is what you used to see on all of the layouts that are now gone)
- **Improved Filter option in the Transcript**
The Filter option on the Transcript uses a tree-structured control that is much easier to work with.
- **RGS can now dynamically change schemas**
Select a new schema through Configure > License Manager, restart the tool and - *viola!* - RGS now includes all commands and objects for the new schema. The schemas you can access are limited by the schemas authorized by your license.
- **New support for progressives**
With the following configuration (after restart), RGS automatically sends a `joinMcast` command for each progressive device that is exposed in the `descriptorList`. RGS then starts sending `setProgressiveValue` commands using multicast - one for each multicast group (progressive device).
 - ◆ Start RGS, and go to **Configure > Engine Options**.
 - ◆ Select the **Automate Multicast Host Behavior** checkbox.
 - ◆ Set **Automatic Progressive Updates Interval** to a non-zero value.
 - ◆ Select the **Automatic Progressive Updates Using Multicast** checkbox.
 - ◆ Click **Apply**, and then click **OK**.
 - ◆ Restart RGS.

RGS 1.x vs. RGS 5.0

We have recently completed a rewrite of our G2S host simulation application. The following provides an overview of the changes and benefits of that rewrite:

1. A Change in Core Technology

- a. Old - Since RadBlue tools need to accommodate so many different extensions to the protocol, our original strategy of using a code generator (such as XMLBeans, or JAXB), while quick and easy to do at first, proved incredibly inconvenient as we had to fork the RGS code with each new schema that we encountered. A code generator can only handle a single schema.
- b. New - In our rewrite, we employ a schema-independent POJO solution, in which we hand-generated a chunk of generic Java code for each command. The net result is that RGS can now handle as many different valid G2S schemas as needed, and can even switch schemas at run-time in a single version of code (more on this later).

2. A New SOAP Stack

- a. In the new RGS, we moved from Apache's Axis 1.4 to Apache's CXF web-service.
Benefits of this change: CXF is a modern, high-performance stack that is easier to use and, most importantly, is still in heavy development (Axis support ended in 2006). CXF has a much better implementation of the web service standards (SOAP and HTTP/REST) than Axis did, resulting in much more stable (and better performing) products.

3. CT - The Common Transport Module

- a. Old - The transport logic in the original RGS product was pretty basic
- b. New - The new RGS incorporates our new Common Transport (CT) module.
Benefits of this change:
 - i. CT is a full and complete implementation of G2S Chapters 1 and 2.
 - ii. CT is an independent module that is capable of G2S host or EGM transport for G2S, so it can be used in many of our products over time.
 - iii. CT handles Point-to-Point and MTP transports, SSL, keepAlives, and the complete communication state machine described in Figure 2.1 (host or EGM side) of the G2S protocol, including full error support. As an example, when we recently encountered a situation where an EGM was sending us a `optionList` while in synch state, we added code to the CT module to return an APX009, and now that change is available in all of our G2S host products.
 - iv. Multicast Transport support is baked into the product, so if you want progressive updates automatically sent using multicast, RGS will now automatically send Join Multicast Group commands to the EGM for all progressive devices, and it will then use those multicast sessions for all subsequent updates.

4. New RGS Business Logic

- a. Near fidelity with the old RGS.
We were able to port all algorithms, databases and user interfaces (most of which were incorporated into the enhanced transcript).

- b.** The new business logic adds multi-EGM support (limited to 5 EGMs), which should make it easy to compare messages from multiple EGMs, or an G2S EGM vs. a G2S SMIB, etc. Multiple EGM support is extended to the progressive, voucher, and other databases as well.
 - c.** The data repository is gone.
RGS now has a full data model for each EGM that is used for Send Commands. Command objects are now integrated in the new Transcript, so there's no limit on the number of command instances that can be reviewed.
 - d.** The event summary panel is gone.
The new Transcript provides the same functionality in one location (Quick Filter on eventReport, view the new summary that shows the eventId and eventText, and then the eventReport command object is available when you double click on a row.
 - e.** RGS can now dynamically change schemas.
Select a new schema, restart the tool to update the user interface, and you see that the tool now includes all commands and objects for the new schema. The schemas you can access are limited by the tool (and currently, by installer). Our new RGS installer automatically installs the schemas authorized by your license.
- 5. Custom Configuration Module (Optional)**
- a. Scriptable Startup Sequence**
 - i.** Old - In the old RGS, the start-up algorithm was baked into the product so it was very difficult to do any start-up algorithm configuration other than selecting or deselecting command execution.
 - ii.** New - In the new RGS, we've scripted the G2S start-up sequence so we will be able to easily mimic the start-up sequences used by different host vendors, including errors and timing. The new start-up engine will also allow additional customization such as the reordering of commands, grouping multiple commands together in one G2S message, sending custom messages, etc. The easy to use GUI means that anyone should be able to create (or update) a host start-up sequence.
 - b. Response Manager**

You can now specify custom responses for any command (changes are applied when saved).

 - i.** "Evil" and third-party commands are supported (specify a custom command).
 - ii.** Optional validation will allow custom scripts to send third-party additions to G2S commands and even invalid values for attributes.
 - iii.** Specify "no response" for any command.
 - iv.** Specify a G2S error or a custom application error for any command.
 - v.** Set up response patterns that are used by the tool (for example, use a standard response for three messages, use a custom response for the next two messages, and then repeat the pattern).

c. Script Runner

The script runner provides a wizard-driven utility to automate complex command sequences. In our first offering, we've automated the download process with a single script and you can have the RGS do any (or all) of the following functions:

- Send an `addPackage` command to instruct the EGM to download a new package. The URL, transfer parameters, and other needed bits of information are entered using the wizard, and can be edited as revisions are needed.
- Once the package is downloaded, the script automatically requests a package list and then reads the contents of the newly downloaded package.
- The RGS can optionally request that the EGM authenticate the package using GAT commands, install the package using a `setScript` command, and even authorize the script in order for the install to take place.

In the future, we'll add scripts to automate `commConfig`, `optionConfig`, media display sequences and many others.

6. Remote Control API (Optional)

Rather than having to run scripts, change start-up algorithms, and configure the response manager through the user interface, some teams are interested in automating their regression testing of an EGM. The Remote Control API will allow a test engineer to write a complete application that can remotely control the new RGS product using an HTTP/REST interface. Some of the high level features include:

a. Configuration

Change the current start-up algorithm script to emulate a specific manufacturer's host before starting an extended test. Change the current response manager script to provide the appropriate host response for the next set of commands that will be coming in from the EGM.

b. Remote Marlie Scripts - Send a Marlie script for the RGS to execute - as simple as a single command, or as complex as an extended series. When the RGS completes execution, the EGM's response to all of the commands in the script is returned to your program. Marlie scripting is much simpler than constructing the complete G2S commands, plus you can take advantage of the RGS data model and common Transport engine.

c. Workspace Control - The RGS workspace is similar to the RGS Transcript. Your program can clear or fetch the workspace for a specific EGM to see the G2S commands that were initiated.

Version 5.0 Modifications

New Features

- RGS now supports student licenses.

Improvements

- The Startup Algorithm, previously on the Engine layout, has been moved to a new Custom Configuration layout. From the Startup Algorithm tab you can enable or disable commands that are sent in the startup algorithm. In addition, the following commands have been added:
 - ♦ Responses: `commsOnlineAck` and `commsDisabledAck`
 - ♦ Requests: `setCommState`, `setKeepAlive`, `getDescriptor`, and `getMeterInfo`
- The Engine Options screen on the Configure menu has been updated with consistent wording and a more user-friendly option order.
- The RGS installer has been updated to determine the version type through the license. Now, one RGS installer can be used to install any version of RGS.
- The Windows version of the RGS installer sets the Windows privilege to allow all users to modify the following directories: `conf`, `database`, `logs` `temp` and `license`. This allows a non-privileged user to run the RGS.
- An **Automate Multicast Host Behavior** option has been added to the Engine Options screen.
- The Transcript database now saves user comments.
- Multicast messages have been added to the Filter option screen on the Transcript Control. Also, when RGS receives an unknown G2S command (the command name is not in the `g2s-command-set.xml` file), it is automatically added to the Transcript Filter option (selected, by default). Finally, `eventReport` messages are now included in the Transcript.

Corrections

- In the Transcript Control, the `optionList` command view now looks similar to the RGS floor tab `optionList` in the RGS 1.x version.
- The **From Location** column on the Transcript now displays the correct location.
- Communications class command errors now display in the Transcript.
- RGS now transitions the EGM from the Sync state to the Online state as soon as it sends the `setCommsState` command to the EGM.
- The progressive broadcast interval now updates the Progressive Service (the entity that is sending the progressive updates). The previous interval must finish before the new value is used. If the interval was set to 0 then the user may have to wait 5 seconds to get the interval value.

- An issue in which RGS dropped the entered Key Off Promo Amount in the `handpay.setRemoteKeyOff` command when constructing the message to send to the EGM has been corrected.

The `handpay.setRemoteKeyOff` command is pre-populated with the details of the latest `handpay` command (or retains the entered values). Previously, it always initialized to `transactionId = 0` and all amounts = 0.

- RST now generates a **G2S_IDE103** event when a card is removed, even if the `setIdValidation` command is not received from the host.
- An issue with the **Fail All GAT Verification Results?** option, which appears on the **Configure > Engine Options** screen, has been corrected.
- Previously, the `doVerification` command had a default value of zero (0) for the `endOffset` attribute when it should be negative one (-1). The default value is now "-1."
- In addition, the notation for the `algorithmType` value did not match the G2S protocol specification. This has been corrected.
- The following **setScript for Module** issues have been corrected:
 - ♦ When only a module is entered and selected to install, no error is returned. However, when you click **Save**, nothing happens.
 - ♦ When only a module is entered and selected to uninstall, no error is returned. However, when you click **Save**, nothing is added to the command.
 - ♦ The `pkgId` attribute in the `moduleCmd` command is optional in the protocol, but is required in the interface. Even when the operation is not equal to **G2S_install** it still requires a `pkgId` attribute.