radblue

RGS Version 44 Release Notes

Release Date: 09 December 2014

Release Summary

In this release, we

- Added support for several G2S Manufacturer extensions – 2h (Standalone Progressive Class), 2j (mediaDisplay Host to Content messaging), 2n (EMDI Connection Required support), and 2p (mediaDisplay Content Token support)
- Modified the Response Manager in the Tester Toolkit so it can affect commsOnline and commsDisabled messages, send duplicate responses to a message, and send message level (MSX) errors.
- Added functionality to the transcript so the view message and message compare screens can be easily maximized when you encounter a large G2S message
- Improved the Remote Control (REST) interface to the tool

And, we added a number of smaller enhancements and fixed several bugs that we encountered over the last several months.

We hope you enjoy this new release. If you have any ideas for improvement that we may have missed, just let us know at support@radblue.com.

The following sections provide details on each of the significant modifications and improvements.

DMV Compare Improvement

- The snapshot comparison functionality was not calculating deltas properly when meter values exceeded $1000 (100000000 in millicents). The commas that were included in the displayed values were throwing off the compare engine. All is now fixed.

Engine Modifications

- **The RGS keep-alive-monitor was not accurate** - The RGS keep-alive-monitor logic now reports a missed **keepAlive** at the configured **keepAlive** interval + 2.5 seconds. Once three of these intervals have passed with no message received from the EGM, and then another 2.5 seconds goes by, the EGM is marked as LOST. The RGS will then respond to subsequent commands with a G2S_MSX003 (Communications Not Online) error, until the EGM reestablishes communications using a **commsOnline** command.

Engine Modifications (continued)

- Four Manufacturer Modifications were implemented in the standard version of RGS in release 44. They include extensions to the mediaDisplay class, along with the addition of a new class that supports Standalone Progressives. More complete details of the impact of each of these extensions follow:

Extension	Summary	Impact on RGS
2h	Standalone Progressive Class to audit remote progressive controllers. This was a manufacturer extension (2h) that was later incorporated into G2S 3.0 in the **g2sSP** namespace.	Added support for the Standalone Progressive Class, which is triggered by the presence of a G2S_spc device in the **descriptorList** of the EGM. This new class adds 5 host-originated commands: **getSpcLog**, **getSpcLogStatus**, **getSpcProfile**, **getSpcStatus**, and **setSpcState**. Support was also added for the one EGM initiated command: **spcLevelReset** (response is **spcLevelResetAck**), and all spc class **eventReports**.
2j	Extends the **mediaDisplay** class with 2 new commands in the **g2sHCI** namespace, so the mediaDisplay host can interact with content running in the mediaDisplay device.	Add **mediaDisplay.hostToContentMessage** (notification) which can be sent to the EGM via a mediaDisplay device (Send Command to EGM screen). The entered string for *instructionData* is converted to base64Binary as per section 25.22.2 of G2S 3.0. Accept **mediaDisplay.contentToHostMessage** (notification) from the EGM.
2n	Extends the **mediaDisplay.loadContent** command with 2 new attributes (*emdiConnectionRequired* and *emdiReconnectTimer*) in the **g2sCR** namespace.	Support for load.content.*emdiConnectionRequired* (boolean) and loadContent.*emdiReconnectTimer* (default=30 seconds) were added to the send panel and the tester toolkit in the RGS. On the send command panel, the entered values are persisted in the GUI to facilitate resending the command.
2p	Extends the **mediaDisplay.loadContent** command with 1 new attribute (*mdContentToken)* in the **g2sCCI** namespace.	Support for load.content.*mdContentToken* was added to the send panel and the tester toolkit in the RGS. On the send command panel, the entered value for this attribute is persisted in the GUI to facilitate resending the command.

Transcript Modifications

- The **G2S transcript** was enhanced by adding **Max** and **Restore** buttons to the View Transcript Message window (launched when you double click a message in the transcript). Since these buttons are located in the top frame of the control, they are available in the Command View and the XML View of the message. This feature is most useful when you encounter a really large message.

Max – expands the screen to the full size of the home screen of the computer.

Restore – Restores the window to its previous size and location.

- The same **Max** and **Restore** button functionality was added to the **Compare** button in the transcript. This feature is used to compare two similar messages in the transcript, so it was also a good candidate for a maximize feature.
- The EGM Filter in the **SOAP transcript** did not list any of the individual EGMs whose messages were in the transcript, so it was impossible to filter to just one EGM. This has been corrected.
- Large messages (specifically the **setEventSub** request) were not making it into the SOAP transcript. This issue is now resolved. The Transcripts can all handle G2S messages up to 20MB (which can be expanded if necessary by modifying the .vmoptions file).

Tester Toolkit Modifications

- **Response Manager** - The RGS Response Manager can now be used to suppress RGS responses to commsOnline and commsDisabled requests from the EGM.
- **Response Manager** - The RGS Response Manager can now send message level (MSX) errors as a response to any G2S request from an EGM.
- **Response Manager** - The RGS Response Manager can now be instructed to duplicate responses to the EGM. If **Send Duplicate** is selected (in the Tester Toolkit – Response Manager modification screen), the RGS will duplicate the responses defined by that Response Manager rule.
- **GAT Management Control** – The GAT Management Control in the Tester Toolkit can create a digest of signatures for all software in a reference EGM (**Create GAT Component Digest**), and then can verify that a second EGM generates matching signatures for all software with the same component Id as in the reference machine. The improvement in RGS 44 is that the RGS will now send a correct value for the *passed* attribute value in the gat.verificationResultAck command, so you can test that an EGM properly disables itself when a invalid software signature is encountered.

Remote Control Modifications

This RGS Remote Control module allows your test console to remotely control the Tester Toolkit, access the Transcript, and control the Advanced Transcript Analyzer through an easy-to-use REST interface. In this release, we made the following improvements to the interface:

- You can now remotely set the Active Response Manager file being used by the RGS. Response Manager definition files can be easily created via the RGS User Interface and shared between different instances of the RGS. The Active Response Manager file (the set of rules currently being used by the RGS) could always be set via the UI, but now can also be set programmatically via the Remote Control interface.
- You can now remotely set the Startup Algorithm file being used by the RGS. Any possible combinations of commands can be defined for the RGS Startup Algorithm using the RGS User Interface, and then saved for later use. The Active Startup Algorithm file (the set of rules currently being used by the RGS) could always be set via the UI, but now can also be set programmatically via the Remote Control interface

radblue

RGS Version 42 Release Notes

Release Date: 26 August 2014

Release Summary

In this release, we did quite a bit of work on the data model snapshot comparison utility – revisiting it to clean up a number of items that we discovered during our rewrite of the Quick Start Guide – our primer for G2S and our tools. We also did some tuning of the `setRemoteKeyoff` command – making it easier to use in the send command panel, and adding it to the custom scripting glossary so you can now create scripts that will remotely key off jackpots occurring at a connected EGM.

We hope you enjoy this new release. If you have any ideas for improvement that we may have missed, just let us know at support@radblue.com.

The following sections provide details on each of the significant modifications and improvements.

Installer Modifications

- **All tools have been updated to use Java 1.7.0_65** – To keep up with the latest security (and other) improvements, all tools have been updated to Java 1.7.0_65. When installing this release, the Windows installer will download Java as part of the installation. If you are installing multiple tools on your computer, this Java upgrade is only done once. The Linux installer bundles the JRE in the installer.

Tester Toolkit Modifications

- `handpay.setRemoteKeyoff` has been added to the Custom Scripting engine and will automatically use the transactionId and amounts from the latest handpay request for the targeted EGM.
- An updated set of sample Custom Scripts are now installed by the RGS (if Tester Toolkit is present), so you can see some examples of Custom Scripting:

Script	Description
balanced-meter-example-001	Create snapshot, prompt for EGM activity, create second snapshot, then run the Balanced Meters Report, which verifies that all meter updates are done in accordance with Appendix B
media-display-example-001	Loads content, sets it as active, show the mediaDisplay, wait for a bit, hide the display, and then release the content.
package-regression-test-001	Add a package, do a GAT verification of that package, and then delete the package. Do this over and over to find memory leaks.

If you don't have this functionality and would like to try it out, just let us know at support@radblue.com...

User Interface Modifications

- The transactionId and amounts in the `handpay.setRemoteKeyoff` control will now automatically default to the values that were sent in the latest handpay request for the selected EGM

DMV Compare modifications

The RGS maintains a copy of each EGM's data model, gathering up and storing all information that is available through the G2S protocol as it is received from the EGM. The data model can be viewed on the Data Model Viewer (DMV) tab on the Databases layout in the RGS – just expand the tree to see each EGM and then you can drill down to the details of any device in the data model.

A powerful feature of the DMV is the ability to snapshot a copy of an EGM's data model, and to compare any two snapshots to review the changes that occurred in the EGM between any two instants in time. Recently, while updating the Quick Start Guide, we did a detailed review of the DMV Compare functions and found a number of important improvements that were made in this release. Specifics on these changes follow:

- The DMV Snapshot Comparison now compares device profile attributes in addition to the option configuration parameters. Using this new functionality, you can create a snapshot, change some parameters, request updated optionLists and deviceProfiles from the EGM, do a second snapshot, and then easily see if the changes made to option parameters correctly flow through to the device profile. (Let us know if you have any questions on this – support@radblue.com.)
- The comparison statement in the DMV Compare report is now a bit friendlier on the screen and on the report: "Comparing RBG_1234-snap1 [Older Value] to RBG_1234-snap2 [Newer Value]"
- The columns that can be used when Quick Filtering the reports have been improved to use the most relevant columns in each type of comparison (meters, profile, status, etc.)
- The DMV Compare Filter GUIs got some updated labels (these let you reduce the results by filtering out unwanted sets of data).
- Negative percentage values in DMV were not formatted properly (this can occur when comparing the average payback percentages between data model snapshots).
- The DMV Snapshot Comparison now correctly compares wager category, game denomination, and currency meters. A new "Group" column contains the denomination or wager category description making the result screen much easier to use.
- For reporting in this new Group column, Game denominations do not specify a currencyId as they use the currencyId of the cabinet. Currency meters now use the format "5.00 USD" so they are now relevant for all cases.

For more details on using this powerful utility – check out the Advanced Skills module (Module 6) in the Quick Start Guide.

RGS Version 40 Release Notes

Release Date: 28 May 2014

Release Summary

In this release, we added quite a few usability improvements to the transcript control – color-coding the XML documents, letting you save changes that you make to the transcript layout and improving quick filter results so the display now includes the selected record when you clear the filter.

More specifically to the RGS, we added support for the master reset extension (1e), did some work on memory management so the RGS now runs better for extended periods, and added a new Send Custom Message option that allows you to send any G2S message to an EGM and the RGS will fix up the appropriate bits before sending it out the door.

We hope you enjoy this new release. If you have any ideas for improvement that we may have missed, just let us know at support@radblue.com.

The following sections provide details on each of the significant modifications and improvements.

Installer Modifications

- **Player databases are now persisted** – The player related databases (player, informedPlayer, and WAT) are now persisted between installs of RGS.

- **More Complete Error Reporting** – If the MAC address in the license file from RadBlue is not found in the PC, the resulting error message now displays a list of the MAC addresses that are seen by the installer for speedy solutions to installation problems.

- **All tools have been updated to use Java 1.7.0_51** – in an effort to stay current, all tools are being updated to a new version of Java. Java 8 is almost ready for prime-time, so we'll probably move to that new major version in the next release of our products.

Configuration Modifications

- **Configure how unknown IDs are handled** – The RGS has a new (configurable) behavior for the ID Reader Algorithm (**create new player on unknown ID**). If selected, a new player record is automatically added when an unknown ID is received by the RGS. If not selected, a setIdValidation response with idType="G2S_invalid" is returned to the EGM.

  The new option is located here: **Tools > Configure > Engine Options > Message > Player**

Transport and Security Modifications

- **Proper handling of SCEP certificate chains** – The RadBlue SCEP client was not properly handling a certificate chain from an SCEP server. The tool was properly storing the first certificate, but then overwrote that certificate with the second certificate in the chain. Resolution - we now verify that the expected distinguished name matches the certificate before we store the certificate. (This is a common security routine used in all products).

- **Import/export self-signed certificates** – When the self-signed certificate option in the tool is used to create a signed certificate, that certificate has to be shared with other end-points so they trust the connection. New export and import options have been added to the Key Store tab in all tools so you can easily export a certificate from, and import a certificate into the tool's keystore file.

Engine Modifications

- **The GTECH Master Reset (gtkMR) extension (1e) has been implemented in the RGS** – This cabinet class extension is used by the host to direct the EGM to return to its factory settings.

- **Memory Management tune-up** – A complex script created by one of our user groups really blew out the memory utilization the RGS. Extended sessions with memory management tools resulted in a new version of the RGS that can run for extended periods with this complex script – a dramatic improvement. If you've had resource issues with prior versions of the RGS, try release 40 and let us know how it works for you.

- **Session Manager has been overhauled** - The RGS Session Manager was not allowing a second request to be sent to the EGM until the response to the prior request was received or timed out after 30 seconds. Since errors didn't count, the 30 second wait became very obvious. The code has been tweaked so the RGS now properly sends G2S requests asynchronously, so you can send a second request immediately.

User Interface Modifications

- **New "Clear Lost EGMs" button on the Engine tab** – The RGS doesn't automatically free EGM slots when EGMs disconnect, because the communications disconnect might just be temporary during the run of a script or other similar activity. A new **Clear Lost EGMs** button on the RGS Engine tab will remove all EGMs that are currently not connected from the RGS's internal list of EGMs. This is especially useful when you are switching between multiple EGMs on a single instance of the tool.

- **Send Command Panel bonus.setBonusAward command** – The setBonusAward control (in the Send Command panel) now uses better default values (Bonus Mode: IGT_standardMode, ID Restriction: G2S_none, and ID Reader Type: G2S_none).

- **Send Command Panel gat.doVerification command** – Seed and Salt value constraints have been added for each of the different Algorithm Types in the gat.doVerification command, to ensure entered values are in synch with the G2S protocol document.

- **Send Command Panel gat.doVerification command** – Salt values (used in the GAT.doVerification command) can now be entered in hex or as a text string.

- **Send Raw XML panel was not retaining its size and location** – but now it does…

- **Player Database – Loss Limit is now in decimal format** – In the Player Information screen of the Player Editor, the loss limit value is now expressed in decimal format rather than millicents.

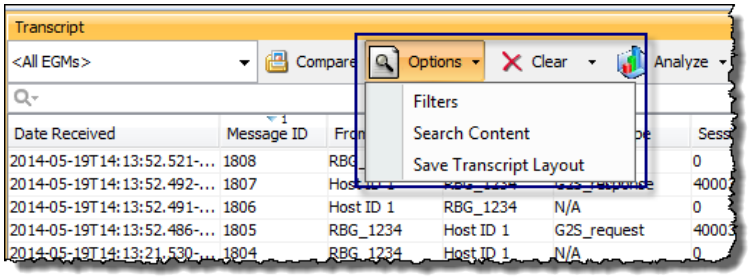
- **A new "Send Custom Message" control has been added** – While Send Raw XML allows you to send anything to an EGM, with the new **Send Custom Message** control, you can send any G2S **request** or **response**, and the RGS will automatically "fix" any of the following fields (so you can send a particular request (or response) over and over and over again). Just paste any command into the control – remove any attributes as appropriate, and then send the command. Commands are persisted in case you want to resend the same request or response.

  The rules:

Attribute	If the attribute is present	If attribute is missing
g2sBody.hostId	Overwrite with RGS hostId	Add using the RGS hostId
g2sBody.egmId	Overwrite with the selected EGM ID	Add with the selected EGM ID
g2sBody.dateTimeSent	Overwrite with current time	Add using the current time
classElement.deviceId	MUST be present	MUST be present
classElement.dateTime	Overwrite with current time	Add using the current time
classElement.commandId	Overwrite with current command ID	Add the current command ID
classElement.sessionType	Leave as is	Do nothing (defaults to G2S_request)
classElement.sessionId	Leave as is	Add the current session ID if sessionType="G2S_request"
classElement.sessionRetry	Leave as is	Ignore
classElement.timeToLive	Leave as is	Do nothing (defaults to 30 seconds)
classElement.sessionMore	Leave as is	IGNORE
classElement.errorCode	Leave as is	Do nothing (defaults to G2S_none)
classElement.errorText	Leave as is	Do nothing (defaults to empty string)

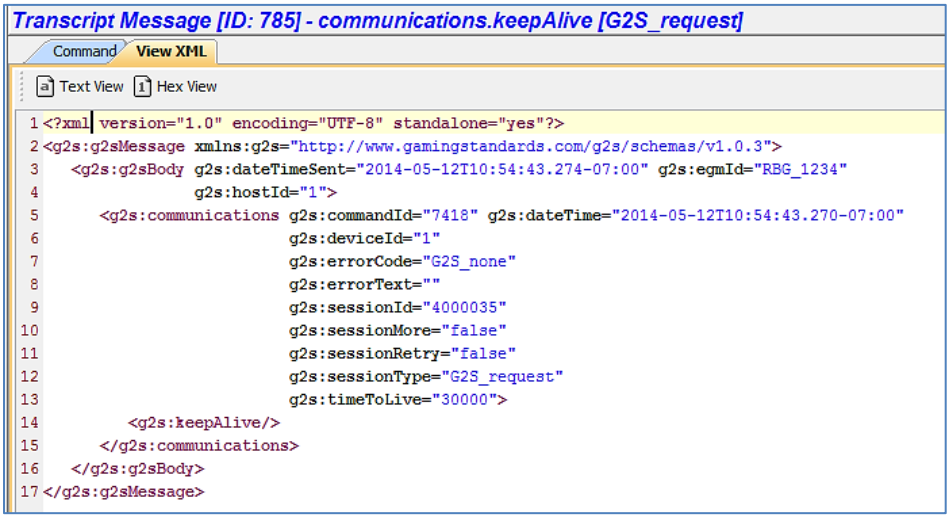
Transcript Modifications

- **Minor reorganization of Transcript Controls** – Since UI changes are usually the most torturous, we'll start the Transcript section with this one. In this new release of the tools, the Search and Filter options have now been grouped with the new Save Transcript Layout option in a new group called **Options:**



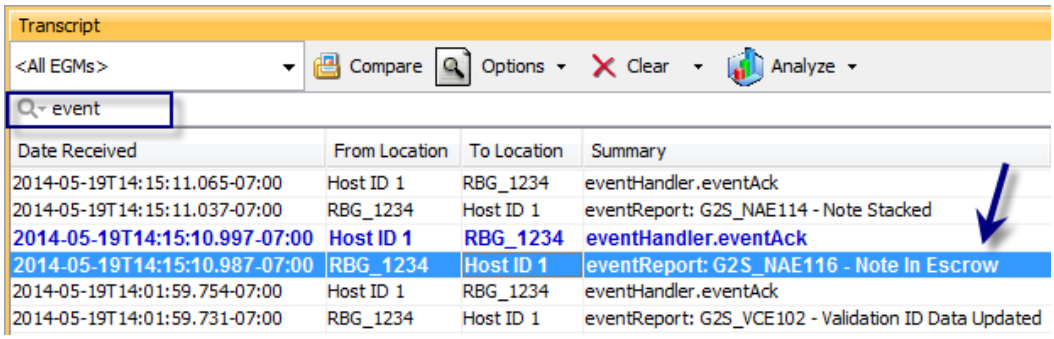
- **Column order, size, and selected columns are now persisted** – Make any changes to the layout of one instance of the transcript, then select **Options** > **Save Transcript Layout** to save your changes as the default layout. This new format is then used for all Transcript instances *in that tool* once you restart.

- **View XML is now color-coded!** We found a cool control that allows us to color code the XML document seen in the View XML object for Transcript Messages, using a unique color for elements, attributes and text strings. This makes the XML MUCH easier to read:



- **Improved reporting when real-time updates is disabled** – When the real-time update of the transcript is disabled, the **Realtime Update** label now goes to Red and Bold, and a message is displayed on the transcript table to make it obvious that the table is no longer being updated in real time.

- **Improved Quick Filter results display** – If you use the Quick Filter control in the transcript to filter the current view based on the entered string, and then select a specific message in the resulting transcript set, that record will now be in the active window when you clear the filter (so you can easily see the G2S messages around the selected message).



- **No more "Not Set" tool tips** – In the Transcript command view, there were lots of instances of tool tips that displayed as "Not Set" (because they weren't defined). These undefined tool tips are no longer displayed.

- **More readable denominations** – In the command view for the meterInfo command, denominations are now shown in decimal form, instead of millicents for currency meters and game denomination meters.

Tester Toolkit Modifications

- **GAT Management Control enforces Seed and Salt value constraints** – In the automated GAT Management Control in the Tester Toolkit, Seed and Salt value constraints have been added for each of the different Algorithm Types in the gat.doVerification command, to ensure entered values are in synch with the G2S protocol document. This will be noticed when building a digest of the signatures of all installed software components in an EGM (or when verifying against a signature digest that you have created).

- **GAT Management Control usability improvement** – In the Verify Components screen of the automated GAT Management Control in the Tester Toolkit, you can now select multiple software components that are to be verified by the tool. The selection can be for a range (shift select) or single instances (control select).

RGS Version 38 Release Notes

Release Date: 04 MAR 2014

Release Summary

In this release, we added a GAT verification tool to the Tester Toolkit, the ability to export the Balanced Meters Analysis Report to an Excel spreadsheet from the Tester Toolkit's Custom Scripting and several configuration parameters to the backup feature. In addition, we corrected several issues.

New Features

- When you run the **Balanced Meters** script verb in **Custom Scripting**, you can now export the **Balanced Meters Analysis Report** to an Excel spreadsheet by clicking **Export to Excel** on the Balanced Meters tab (**Tester Toolkit** > **Custom Scripting**). To export errors only, select the **Export errors only** option before you click **Export to Excel**.
- A new **GAT Management Control** feature has been added to the Tester Toolkit layout that can be used as a sample GAT software verification tool. The GAT Management Control consists of two parts: the **GAT Component Digest** and the **GAT Component Verification**.

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

  To create a new digest file:

  a. Select **New**, and enter a file name and description for the new digest file.
  b. Click **Create New Component Digest File**. A blank digest file is created.
  c. Select a **Connected EGM** that will be used as the reference instance
  d. Select **Verify Components**, which launches a new object that contains a listing of all components in the selected EGM, as were reported by the EGM during the start-up algorithm.
  e. Select a row, and click **GAT doVerification** to have RGS instruct the EGM to verify the selected component using the provided values.

     or

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

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

As each component–algorithm combination is verified and a result is returned by the EGM, the row changes to light purple shading, indicating that this value can be added to the digest file.

f. Once all selected rows have been verified, select **Add to Digest** to move this set of component verification results into the digest file for later use. This set now forms the digest of the software signatures of the known-good software on the control EGM.

GAT Component Verification

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

To verify the components against a GAT Component Digest file:

a. Select the **Component Digest File** that contains the known-good software results. If a newly created file is not in the drop-down list, press the **Refresh** to reload the file list.

b. Select an EGM that is connected to the RGS. When the EGM is selected, the **GAT Components** table is populated with a listing of each of the software components in the EGM. The RGS then randomly selects an algorithm for each component from the set of combinations that are common between the EGM under test and the GAT Component Digest File.

Note: To edit the verification record of a component, double-click the row containing the component you need to change. Then, select an alternate **Algorithm Type**, or exclude this component from the verification.

c. Once you are satisfied with your selections, press **Start Verification** to begin the verification process of the selected components.

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

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

Installation Modifications

The following configuration parameters are now retained between installations:

- Active Startup Algorithm (**Tester Toolkit** > **Startup Algorithms** > **Active Algorithm**)
- On **Tools** > **Configure** > **Engine Options** > **Filters** screen:
  - Filter G2S ACKS from Transcript
  - Filter G2S Keep Alives from Transcript
  - Filter G2S Set Progressive Values from Transcript
  - Filter Multicast G2S Bonus Activities from Transcript
- On **Tools** > **Configure** > **Engine Options** > **Database** screen:
  - Clear On Startup
  - Enable General Transcript Analyzer

Configuration Modifications

A new **Negotiate Namespace Behavior** option has been added to the **Communications** tab, located under **Tools** > **Configure** > **Engine Options** > **Messages**. This option determines how RGS sets the *negotiateNamespaces* attribute in the `communications.commsOnLineAck` command. You can choose to **Never Negotiate** namespaces, **Always Negotiate** namespaces, or select **G2S 2.1 Behavior**, which uses namespace negotiation *only if* it is supported by the EGM. The default is **G2S 2.1 Behavior**.

Engine Modifications

- An issue in which RGS did not properly support a `gat.gatProfile` command when the *supportsSpecialFunctions* attribute was set to **false** has been corrected.

User Interface Modifications

- An issue in which, when using the **setOptionChange** user interface (**Send Command** > **G2S_ optionConfig**), the same nodes appeared in two separate devices (or option sets) has been corrected.
- The getEventHandlerLog control was using the Last Sequence and Total Entries values from the mast recent `eventHandlerLogStatus` command. The control now remembers the last values entered, which works much better.

Transcript Modifications

- An issue in which the data displayed incorrectly in the `eventHandler.eventHandlerLogList` details screen (accessed by double-clicking the message row in the Message Transcript) has been corrected.
- The **Load** option has been removed from the **Multicast Transcript** and **SOAP Transcript** objects because it is obsolete.