Release Date: 07 December 2015

★ New Feature ⬆ Enhancement/Upgrade ⚙ Fix

★ **10521** All tools have been updated to use **Java 1.8.0_60** to keep up with the latest Java enhancements, including security improvements. Java Runtime is bundled with the installer.

★ **10685** With release 52, we are starting to implement the changes for G2S 3.0 on a chapter-by-chapter basis. In this release, we implemented the changes for Chapters 1 and 2.

As we implement G2S 3.0, some classes, commands and attributes will be deprecated.

- When a class is deprecated, the tool will no longer respond and will send **`G2S_APX007 Class Not Supported`** error.
- When a command has been deprecated, the tool will no longer respond and will send a **`G2S_APX008 Command Not Supported`** error.

Once an attribute has been deprecated, RGS will no longer implement the logic associated with the attribute or send the attribute in the XML. However, the command view in the transcript will show the attribute and indicate that it has been deprecated.

Our philosophy is to alert users to G2S 3.0 changes as early as possible. Code that is dependent on G2S 2.1 features should break so that fixes can be implemented.

⬆ **10688** Communication channels (Chapter 2) have been updated for G2S 3.0.

RGS will no longer negotiate namespaces with an EGM that requests name negotiations.

The following items were removed from RGS:

- **`useDefaultConfig`** and **`multicastSupported`** attributes from the **`commsProfile`** command
- **`getNamespaceList/namespaceList`** commands
- **`closeComms/closeCommsAck`** commands
- **`getClassDescriptors/classDescriptorList`** commands
- **`negotiateNamespace`**, **`deviceStateChanged`** and **`deviceAccessChanged`** attributes in **`commsOnLine`** command
- **`configDateTime`** attribute in **`commsStatus`**, **`commsProfile`** and **`descriptorList`** commands
- **`includeActive`** and **`includeInactive`** attributes in **`getDescriptor`** command
- removed content from **`G2S_CME005`** and **`G2S_CME006`**

- `underlyingTransportState` and `derivedTransportState` attributes in `commsStatus`
- `listStateDateTime` attribute in `descriptorList` commands
- `G2S_CME130` - `G2S_CME141` events
- `G2S_CMX011` error

⚙ **10714** When the **Send My Command** script verb in the Tester Toolkit executes, it now honors the **Ignore All Errors** attribute, allowing scripts to send messages to an invalid device. Previously, RGS would report an error if the target device did not exist.

⬆ **10723** Improved transcript display by removing the overlay message indicating that real-time updates are disabled. When real-time updates are disabled, the user interface will display the option in red text.

⬆ **10780** Modified Remote Control to allow script execution in the `sync` state. Previously, RGS reported an error if the EGM was not in the `online` state.

⚙ **10785** Corrected an installer issue in which missing files would not allow RGS to determine the schema versions available for use.

⚙ **10802** Modified the transport service to use the selected schema when creating virtual EGMs. Previously, RGS was using the default G2S 3.0 schema when creating EGMs even though a different schema was selected.

⚙ **10803** Corrected an error in which the schema displayed under **Engine Options** > **General** was not the selected schema.

⚙ **10804** Corrected an error that occurred when a player record did not have valid data. As a result, the player database editor did not save changes.

Release Date: 09 September 2015

Release Summary

In this release of the RGS, we made the following improvements:

- **Upgraded to G2S 3.0** – The RGS now uses a backwards compatible G2S 3.0 schema. We've done extensive testing to ensure that the tool continues to work with G2S 1.1 implementations, but please let us know if you encounter any issues and we'll get them resolved without delay.
- **Added support for the Employee Class** – The RGS now supports the G2S 3.0 Employee class and adds a new employee tab to the Player Editor for Player Database records (so you can specify valid Activity Groups and PINs)
- **Added support for the dft Class** – The RGS now supports the G2S 3.0 dft class when a G2S_dft device is included in the `descriptorList` from the EGM.

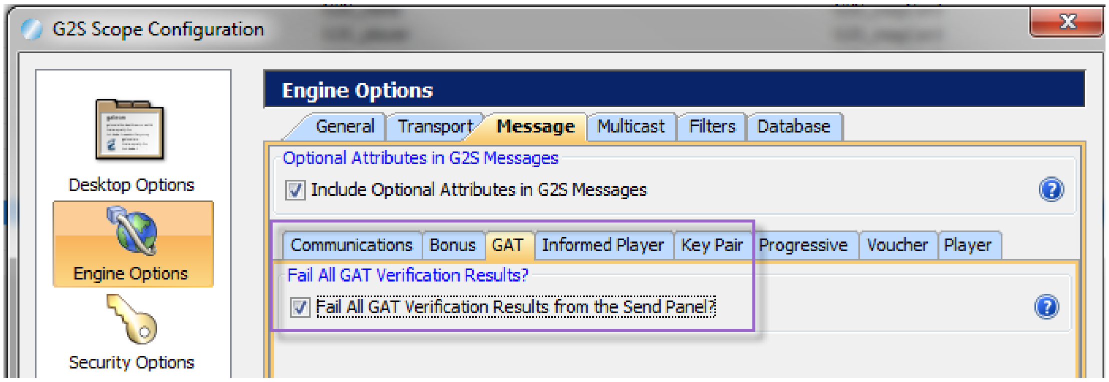
We also added a number of smaller enhancements and fixed several bugs that we encountered over the last several months.

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.

Configuration Modifications

- **Fail All GAT Verification Results is working again** – When implementing the GAT Signature Digest code in Release 48, the configuration option that directed the RGS to fail all verification results from the EGM stopped working. This has been corrected, and the configuration prompt has been adjusted to indicate that it only affects `doVerification` requests sent from the Send Command panel:



Optional Attributes in G2S Messages

☑ Include Optional Attributes in G2S Messages

Fail All GAT Verification Results?

☑ Fail All GAT Verification Results from the Send Panel?

Engine Modifications

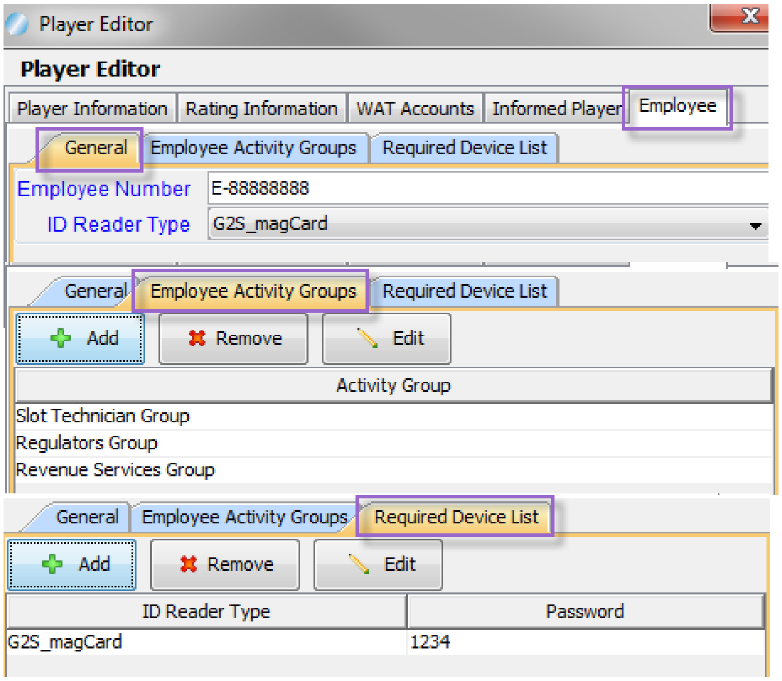
- RGS now uses the **G2S 3.0 schema**, and will validate all messages against that schema (which should be backwards compatible with G2S 1.1). New functionality is based on the G2S 3.0 draft 30 specification from GSA.
- Added support for all commands in the **G2S employee class**, as defined in G2S 3.0. This functionality is exposed when the EGM includes a **G2S_employee** device in its `descriptorList`. New employee fields were added to the Player Database, as this is actually an ID database (record lookup based on the ID number read by one of the idReaders on the EGM).
- Added support for all commands in the **G2S dft class**, as defined in G2S 3.0. This functionality is exposed when the EGM includes a **G2S_dft** device in its `descriptorList`.
- The RGS also supports option configuration for any G2S parameters that are exposed through the **dft** or **employee** devices in the `optionConfig.optionList` command.
- **G2S Extension 2r** – Our early implementation of Player Session Meter Deltas (for Greece) has been updated to align with the published Manufacturer Extension 2r (Player Session Meters). The only significant change was to add *egmException* attribute to the `playerSessionEnd` and `playerSessionEndExt` commands.
- **IGT Smart Card Events had an incorrect prefix** – The Smart Card events had the wrong prefix - Changed the prefix from “**IGT_**” to “**G2S_**” for events **G2S_SCE120** through **G2S_SCE125**.

Transcript Modifications

- Added support for all commands in the employee and dft classes. The best view of the employee session meter deltas can be found in the `employeeSessionEnd` command view.

User Interface Modifications

- The `handpay.setRemoteKeyOff` GUI incorrectly limited amounts to $21,000 - The limits for all three amounts (cashable, nonCashable, and promo) are now $999,999,999.99.
- In the RGS, a new **Employee** tab is added to the Player Database > Player Editor, if the ID_type is set to “G2S_employee”. In the sample database, there are two employees with ids of "88888888" and "99999999". When a new G2S_employee ID is entered, or if you change an ID_type to “G2S_employee”, if you then save and re-edit the record, you will see the new **Employee** tab. When the tab is first added, the default Activity Groups are those from the most recent `employeeProfile` received from the EGM.
  - On the **General** sub-tab, you can indicate the Employee Number and ID Reader type for this employee (typically “G2S_magCard”).
  - On the **Employee Activity Groups** sub-tab, you can manage the list of valid Activity Groups for this employee.
  - On the **Required Device List** tab, you can specify the Authentication Device Type and Password (PIN) for this employee.



Employee Number: E-88888888

ID Reader Type: G2S_magCard

Activity Group
Slot Technician Group
Regulators Group
Revenue Services Group

ID Reader Type	Password
G2S_magCard	1234

Tester Toolkit Modifications

- All of the host initiated **employee** class commands are available in the Startup Algorithm and Custom Scripting modules, so it is easy to create extended regression tests of the employee class functionality.
- In the Response Manager, a custom response can be defined for any of the EGM initiated **employee** or **dft** class commands.
- For the **dft** class, it didn't make sense to include all commands in all modules. The following table identifies the dft commands that are supported in each component of the RGS:

dft Command	Send Command	Custom Scripting	Startup Algorithm
Host initiated g2s request commands			
setDftLockout	X		
setDftState	X	X	X
getDftStatus	X	X	X
getDftProfile	X	X	X
requestDftTransfer	X	X	
cancelDftRequest	X		
getDftLogStatus	X	X	X
getDftLog	X	X	X

Extended Discussion – A Brief Overview of the DFT class

The **dft class** (Direct Funds Transfer) manages the transfer of funds directly between an EGM and a host system application, rather than to and from a wagering account. Transfers from the EGM to the host typically go directly to an application on the system, such as to purchase a keno ticket or to pay an entry fee for a tournament. Transfers from the host to the EGM typically come directly from an application on the system, such as redemption of a winning keno ticket or payment of tournament prizes. The dft class is a multiple-device class (typically one device per dft host)

A dft transfer is a typical four step G2S transaction that is initiated by a request from the host, as per the following simple message sequence diagram:

EGM Command		Host Command	Interpretive Comment
Host initiated request for transfer			
	←	requestDftTransfer	Please start a transfer of this amount of this type of money to or from the EGM
dftRequestPending	→		OK – will do
EGM initiated transfer			
initiateDftTransfer	→		Initiate a transfer of this amount of this type of money in this direction
	←	authorizeDftTransfer	OK - this transfer is authorized
commitDftTransfer	→		The transfer is complete – here are the results
	←	commitDftTransferAck	Results have been received and posted

Extended Discussion – A Brief Overview of the Employee class

The employee class is used to track the actions of employees at an EGM. Like the player class, the employee class is a single-device class.

Meter Reporting – The employee class is similar to the player class, in that it uses session delta records to track meter movements at an EGM. However, the set of meters used for employee tracking is much broader than the set used for player tracking as it includes all device meters changed while the employee was present at the EGM. It does not include **class meters**, wager category, game denomination, or currency meters. Meter deltas are stored in the employeeLog record for the session, and are reported in employee session update events and in the employeeSessionEnd command that concludes each session.

Employee Activity Reporting – The employee class also includes features that can be used by employees to report their activities while at an EGM. For example, an employee can report the reason for entering an EGM, problems discovered, problems resolved, player complaints, etc. A set of property-specific activity codes can be set in the profile of the employee device, and is then filtered by the set of Activity Group codes that are appropriate for each employee (reported by the host in the employeeActivityList response). When this feature is supported by an EGM, an employee can select an activity code from this list to easily report each activity to the host. The authentication device to be used by a specific employee is reported by the host in the `employeeDeviceList`, and the permitted activity groups are identified in the `employeeActivityList`, so the RST will request the current lists for an employee whenever a valid employee ID is inserted into the idReader.

In the RGS, a new **Employee** tab is added to the Player Database > Player Editor, if the ID_type is set to "G2S_employee" (Please see the User Interface Modifications section for more information on this new functionality). The new employee fields in the Player Database record are used by the RGS when replying to Employee Class requests from the EGM.

The following table provides a sample set of G2S message flows for the Employee class:

EGM Command		Host Command	Interpretive Comment (G2S_event)
Employee inserts their ID			
getIdValidation	→		This Id has just been inserted
	←	setIdValidation	It’s an employee Id (idType="G2S_employee") (IDE101)
employeeSessionStart	→		Starting an employee session for this person (EME101)
	←	employeeSessionStartAck	OK – thanks for the information
getEmployeeActivities	→		What are the valid employee activity groups for this person?
	←	employeeActivityList	Here they are (from the Employee – Employee Activity Groups tab in the player database for this ID)
getEmployeeDevices	→		Which device is to be used to authenticate this person?
	←	employeeDeviceList	Use this one (from Employee – Required Device List in the player database for this ID)
An Activity is selected, a valid PIN is entered, then the activity is reported by the EGM			
validateEmployeePIN	→		This PIN was entered
	←	validateEmployeeAck	It is/is not valid
employeeActivity	→		This activity occurred (EME201)
	←	employeeActivityAck	Activity has been received and posted (EME202)
Employee inserts a note, plays a game, cashes out, etc.			
G2S_EME102	→		Employee Session Updated event conveys current delta values (stored in translog). Sent for every update.
Employee removes their ID			
employeeSessionEnd	→		Employee Session is concluded – here are the non-zero meter deltas that occurred during the session. (IDE103 & EME103)
	←	employeeSessionEndAck	Activity has been received and posted (EME104)

Release Date: 03 June 2015

Release Summary

In this release of the RGS, we upgraded to a current version of Java (1.7.0_76), now bundling it in the RadBlue installer. We also made a number of enhancements and corrections to the tool:

- **Remote Control (REST) interface improvements** - The RGS Remote Control Scratchpad now shows the details of the sent and received command
- **SCEP code improvements** – In the last release, we moved to a new transport stack which caused the SCEP code to stop working in some cases. If you are having SCEP issues – this new release should correct them. If not, let us know (support@radblue.com).
- **Continued improvements to the RGS Session Manager** – We found (and fixed) another instance where the RGS would hang for 30 seconds if the EGM did not send a G2S application response to a request sent from the RGS. This was only happening when the RGS requested additional parameters for the command to be sent.
- **Custom Scripting tweaks** – In the Custom Scripting engine (part of the RGS Tester Toolkit), the `setRemoteKeyOff` verb was not allowing all 10 keyoff types. We also made some improvements to the sample Package Regression Script and the Media Display Example Script.
- **Time Zone Offsets** – New controls were added to the Send Command panel that let you set and request Time Zone Offset information in the EGM's cabinet class.
- **GZIP improvements** – A new column has been added to the SOAP transcript to indicate which messages are GZIP encoded. Several minor improvements were made to the RGS's GZIP implementation.

We also 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.

Installer Modifications

- **Updated our tools to use Java 1.7.0_76** – To keep reasonably close to the leading edge of Java development, we've moved our tools to version 1.7.0_76 of Java (we're not quite ready to make the leap to Java 8 in the core tools). This way, we take advantage of functional and security improvements that are made in the latest versions.
- **The Java runtime installer is now bundled with the RadBlue tool installer** – The installer for the requisite version of Java is now included with the RadBlue tool installer. This makes the RadBlue installer a bit larger, but should delight those who are installing our tools on computers that are not connected to the public Internet.

Transport & Security Modifications

- **SCEP code improvements** – In the last release, we moved to a new transport stack which caused the SCEP code to stop working in some cases (an untrapped error was occurring that caused the operation to silently fail). If you are having SCEP issues – this new release should correct them. If not, let us know (support@radblue.com).
- **GZIP processing has been improved** – In prior versions, the RGS would always advertise that all three GSA transport options were available, regardless of the current settings on the Configure > Engine Options > Transport page. The code was improved so the configured selections for GZIP_in_HTTP_STACK and/or GZIP_PAYLOAD are now correctly reflected in the Transport Options calls. A new column in the SOAP transcript now displays the content encoding (blank or "gzip").

Engine Modifications

- **Corrected error sent when incorrect sessionType is received** – The RGS will now send back a G2S_APX008 error instead of G2S_APX009 when command is sent with the wrong *sessionType* (as an example, a `cabinetStatus` sent with *sessionType=*"G2S_request").
- **Continued improvements to the RGS Session Manager** – We found (and fixed) another instance where the RGS would hang for 30 seconds if the EGM did not send a G2S application response to a request sent from the RGS. This was only happening when the RGS requested additional parameters for the command to be sent.

User Interface Modifications

- **Player WAT Balances are now always current** – A successful host initiated WAT transfer to or from the EGM was not updating the Player's WAT balances in the Player database control (though the underlying data model was correct). All balances are now updated when a successful transfer takes place.

User Interface Modifications (continued)

- **Added Support for Time Zone Offsets (cabinet class)** – The RGS Send Panel now supports the G2S commands for setting and retrieving Time Zone Offset information in the EGM's cabinet class. The set command uses a custom control that allows you to set one or more future Time Zone change entries (+/- hours from UTC).

  Note: When using the set command with the RST, be aware that the RST will act on changes that are currently in effect, but will not implement timezone changes scheduled in the future.

- A new **Send g2sAck Error** button has been added to the Send Command panel. This new control allows you to select a command, and then instruct the RGS to cause a transport error when the command is received (Send an MSX error, send no g2sAck, or cause a SOAP fault). Please see the extended discussion of this functionality later in these release notes.

Transcript Modifications

- **New "Content Encoding" column on the SOAP Transcript** – A new **Content Encoding** column was added to the SOAP transcript, and indicates how a G2S message is encoded (for now, it is blank or indicates **gzip**).
- **Transcript Message Viewer buttons not working** – The new Maximize/Restore code added in a recent release caused the **g2sAck**, **REQ-RESP**, and **Transport Message** buttons to stop working. These are used to easily see a message related to the one being displayed, and are now working again.

Tester Toolkit Modifications

- **setRemoteKeyoff verb** – In the Custom Scripting engine, the `setRemoteKeyOff` verb was only displaying 4 keyoff types. This has been corrected so that any of the 10 possible keyoff types can now be selected.
- **Package Regression Test Script** – the `download.doVerification` verb format was recently changed, so we had to redo the Package Regression Test Script so it would use the new verb. If you're using this verb in your scripts, you'll need to change them, as well.
- **Media Display Example Script** – This script instructs a mediaDisplay device to load a new piece of content, then activate it, show it, hide it, and finally release the content. To make it easier to see each step at the EGM, a 5 second pause was added between steps.
- **Response Manager Changes –** The following changes were made to the RGS Response Manager (which controls how the RGS will respond to selected commands):
  - ➢ The option to send a Message Error was removed (MSX errors were incorrectly being sent as application responses, but should be sent in a g2sAck).
  - ➢ The functionality to send a custom response to an eventReport is now working correctly.

Remote Control Modifications

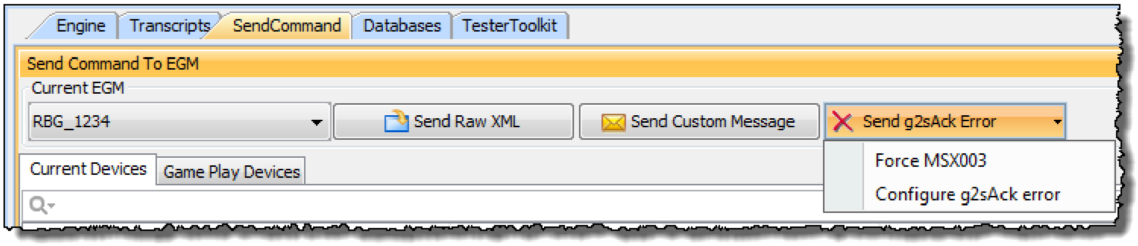
This RGS Remote Control module allows your test console to remotely control the Tester Toolkit (Run custom scripts, change the startup algorithm and the active response manager script), and manage the Transcript (set a marker and request a set of transcript records for analysis) via an easy-to-use REST interface. In this release, we made the following improvements to the interface:

- **Remote Control (REST) interface improvements** - The RGS Remote Control Scratch Pad now shows the details of the request (Method, Path, and payload) as well as the Response returned from the EGM. The scratchpad will be used from this point forward for providing the latest details of all RGS Remote Control calls.
- **Disable Text was being stripped out of setDeviceState commands** – If a custom script containing a `setDeviceState` command was run directly, the resulting command was as expected. However, if that same custom script was run via the remote control interface, the disable text was stripped out of the `setDeviceState` command. This has been corrected.

Extended Discussion – Generating transport errors in the RGS

A new **Send g2sAck Error** button has been added to the Send Command panel. This new control allows you to direct the RGS to create a transport error (no g2sAck, MSX error, or SOAP fault) when a selected command is received by the RGS. This section provides extended information on this new control.

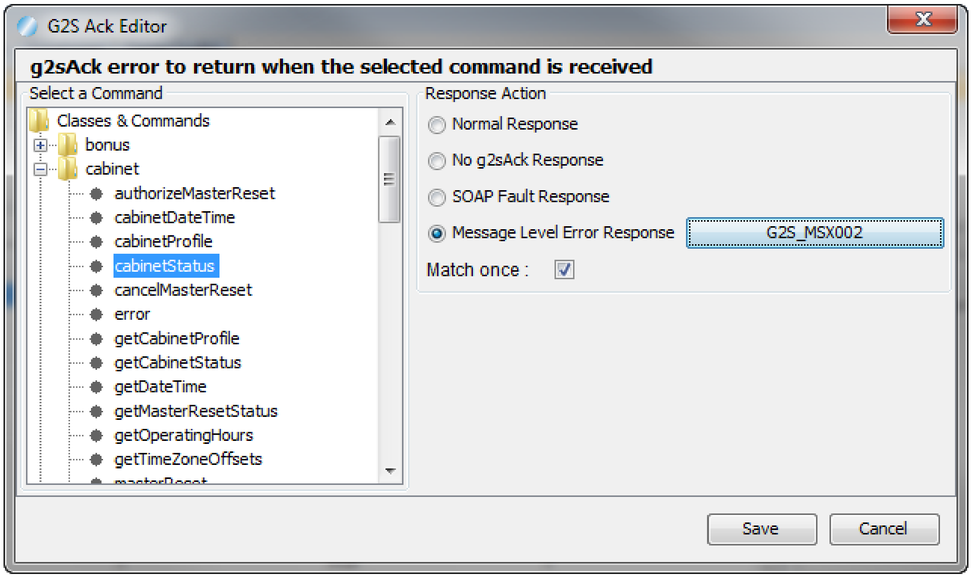
1) When you select the new **Send g2sAck Error** button on the Send Command panel, you are presented with two choices:



   a. **Force MSX003** – replicates the prior functionality, in which the RGS data model for this EGM is cleared, so the RGS no longer knows anything about the EGM. As a result, the response to any command from the EGM will be an MSX003 until the EGM reestablishes communications.

   b. **Configure g2sAck error** – provides the functionality by which a specific command can be selected and the RGS can be instructed to respond with a g2sAck or other transport error when the selected command is received from the EGM. A new selection replaces the prior selection.

2) When **Configure g2sAck error** is selected, the following **G2S Ack Editor** screen is presented:



G2S Ack Editor

g2sAck error to return when the selected command is received

Select a Command

- Classes & Commands
  - bonus
  - cabinet
    - authorizeMasterReset
    - cabinetDateTime
    - cabinetProfile
    - cabinetStatus
    - cancelMasterReset
    - error
    - getCabinetProfile
    - getCabinetStatus
    - getDateTime
    - getMasterResetStatus
    - getOperatingHours
    - getTimeZoneOffsets
    - masterReset

Response Action

○ Normal Response

○ No g2sAck Response

○ SOAP Fault Response

● Message Level Error Response G2S_MSX002

Match once : ☑

a. Using the command selection tree on the left, expand the appropriate class, and then select a command to target.

   Note: Only one target command is permitted in the RGS. Selecting a second command will cause that to be the one target command.

b. Once a command is selected, the Response Action options on the right become active.

   i. **Normal Response** – send a normal response to this command. This is used to easily clear a continuous error (see below)

   ii. **No g2sAck Reponse** – To allow G2S resend testing, the RGS can be instructed to not send a g2sAck transport response to the selected command.

   iii. **SOAP Fault Response** – A SOAP fault will be generated when the selected command is received from the EGM.

   iv. **Message Level Error Response** – When this option is selected, you can then select any MSX error to be returned to the EGM when the selected command is received.

   Note: An MSX003 sent via this mechanism *should* cause the EGM to send a `commsOnline` command, but will not clear the RGS data model for that EGM (see **Force MSX003** above).

   v. **Match once checkbox** – This option can be used with any of the above options (ii through iv) to cause the error to be generated one time, or continuously (whenever the targeted command is received from the EGM). Select **Normal Response** (i above) to clear the continuous error.

radblue

RGS Version 46 Release Notes

Release Date: 11 March 2015

Release Summary

In this release, we

- **Added support for another G2S Manufacturer extension** – Player Session Deltas is used by the host (in Greece) to set a separate meter subscription that allows more granular tracking of what occurred during the player's session at the EGM. If you are playing in that market, please talk to your system vendor for further details.
- **Released a 64-bit Version of RGS** – Some complex EGMs (those with lots of gamePlay devices) may cause the RGS to run out of memory, or to encounter memory management issues due to the size of the G2S data model. We are now shipping a 64 bit version that should resolve this issue, as more memory can be allocated for the tool.
- **Continued improvements in the Remote Control (REST) interface to the RGS**. Folks are finding it useful to be able to remotely drive the RGS from an external program – the more advanced their interactions become, the more things we find to improve!

We also 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.

Engine Modifications

- **Added Support for Player Session Deltas** – So that EGM code can be exercised in advance of interoperability testing for the Greek market, the following modifications were made to the RGS to support the new Player Session Delta extension:
    - **Add support for meterDelta subscriptions** (set, get, display, and added support for meter-deltas in the data model viewer (DMV))
    - **Add support for the playerMeterLog** (get, display, and added support to the DMV)
    - **Add support for the playerSessionEndExt command** (proper responses, with enhancements to the Response Manager so custom responses to the command can be generated)

User Interface Modifications

- **Send Panel – setOptionChange** – To minimize confusion, the setOptionChange GUI now contains a notice to users to indicate that the data I the control is based on the most recent optionList response. If you make parameter changes, you must request a new optionList for those changes to be seen in this dynamic control.
- **Send Panel – setMeterSub** – this control was allowing an invalid periodic interval. The Periodic Interval field now defaults to a minimum value of 60 seconds.
- **Send Panel – setActiveDenoms** – this control was not correctly interpreting the denomination range max value reported by the EGM. Most people don't use denomination ranges, but when they do, this control will now work much better for them!

Installer Modifications

- **64 Bit installer is now available** – Some complex EGMs (those with lots of gamePlay devices) may cause the RGS to run out of memory, or to encounter memory management issues due to the size of the G2S data model. We are now shipping a 64 bit version that should resolve this issue, as more memory can be allocated for the tool.
- **RGS Memory Size Limits are now configurable** – In further support of EGMs with complex G2S data models, the initial and maximum memory size used by the tool is now configurable (Tools > Configure > Memory Options). In general, most will not have to use these options, but if you're having issues, RadBlue might direct you to change these options, and this is an easy way to effect those changes.

GAT Functionality Modifications

- **GAT Management Control Improvements** – 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.

  In this release, the GAT Management Control will now send a doVerification command for all components in the EGM, and can now properly interpret gat.verificationResult commands from the EGM that contain a list of components in a variety of states. For proper processing, the control requires the EGM to properly send a G2S_GAE105 [Verification Result Acknowledged and Passed] or a G2S_GAE106 [Verification Result Acknowledged and Failed] with transaction logs for each component being authorized.
- **Tester Toolkit – Custom Scripting – gat.doVerification verb** now supports a list of multiple components to be verified. Since a script can be used against any EGM, the drop down list of componentIds is not populated, but an instance of the verb can now be constructed that will send a single doVerification request for all components in the EGM – just as is done in the Send Command panel.

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 directly set a Transcript Marker record through the remote control interface. Using this new call, you can to set a Marker, run a script, set a second Marker, and then request all transcript records between the two marker records for analysis by your application.
- One of the teams using the RGS Remote Control interface noticed that the Remote Control engine lost track of an EGM if it restarted too quickly without sending a commsClosing command (in the case where an EGM is powered off, then it comes back online within 90 seconds). The RGS now handles this situation much more elegantly.