



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, RLT 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.

The EGM will no longer attempt to negotiate namespaces with the host. A G2S 2.1 host should see that the EGM is not requesting namespace negotiation.

The following items were removed from RLT:

- **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

The following item was added to RLT:

Error code **G2S_CMX012** was added for invalid class or devices in the **getDescriptor** command



10717 RLT now supports inbound **GZIP** messages. Previously, RLT only supported the compressing of outbound messages to the host.



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.



10724 Corrected an error that occurs when the employee device is owned by the EGM. Previously, the EGM was requesting the employee devices even though there was no host to respond to the message.



10726 Corrected an error that was generated when the maximum configuration ID for the handpay device was set too low. The configuration ID now allows up to the maximum value.



10738 Corrected an error that occurred when the DFT device was owned by the EGM. For EGM-owned devices, the host-enabled flag is now set to **true** on restart. This prevents required-for-play devices from disabling the EGM.



10782 **GSATB-128** has been implemented. Previously, the EGM would reject the same active content ID enabled in different media display devices. Now, the same content ID can be active for different media display devices.

Release Summary

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

- **Upgraded to G2S 3.0** – The RLT 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 RLT now supports the G2S 3.0 Employee class. In the RLT, you can insert an employee card, and then generate revenue and gamePlay activity that will be tracked as employee deltas.
- **Added support for the dft Class** – The RLT now supports the G2S 3.0 dft class. When the host initiates a dft request, the SmartEGM will process the dft transfer, as long as there's a valid G2S_dft device in the data model.
- A new SmartEGM Template file (RadBlue SmartEGM w / dft+employee) is included with this new version of the RLT. This file includes sample G2S_employee and G2S_dft devices (with sample values in the standard configuration parameters).
- **Start Individual EGMs when Engine is Stopped** – Starting the RLT Engine causes all of the EGMs to enter the OPENING mode, so they all start sending commsOnline commands to the host. In this release, you can now start individual EGMs, without starting the engine in order to do a limited test of the host (or a script).

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 for each of the significant modifications and improvements:

SmartEGM Modifications

- **Added support for the dft Class** – A new SmartEGM Template file (**RadBlue SmartEGM w / dft+employee**) contains a **dft device** with a sample set of data for the standard optionConfig parameters.

When the G2S_dft device is present in the SmartEGM's data model, the SmartEGM will now support dft transfers to (or from) the EGM. Please note, if the host requests a dft transfer FROM the EGM and the transfer amount is greater than the current credit meter, the SmartEGM will patiently wait until the credit meter is greater than or equal to the amount of the transfer and will then initiate the transfer to the host. Of course, the pending dft request can always be cancelled via a `cancelDftRequest` from the host.

- Added support for the employee Class** – A new SmartEGM Template file (**RadBlue SmartEGM w / dft+employee**) contains one employee device with a sample set of data for the standard optionConfig parameters. When a valid employee Id is inserted into the Id reader used to validate employees, the SmartEGM will send the following G2S requests to the host:

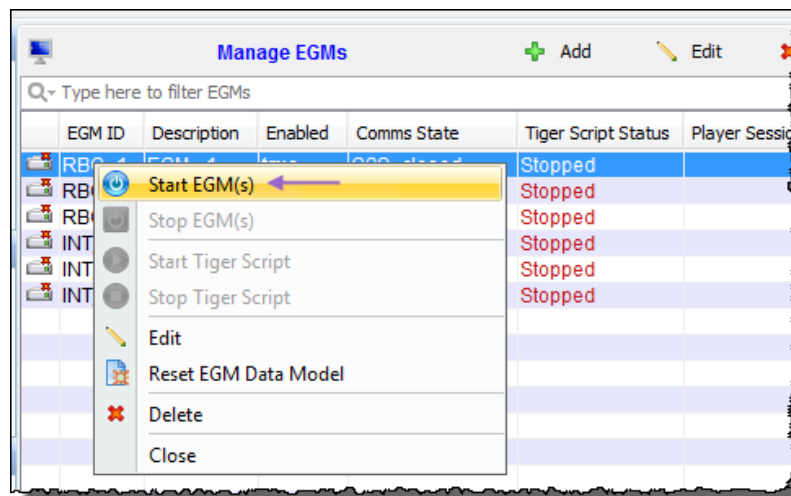
EGM Command	Host Command	Interpretive Comment (G2S_event)
Employee inserts their ID		
getIdValidation	→	This Id has just been inserted
	←	It's an employee Id (idType="G2S_employee") (IDE101)
employeeSessionStart	→	Starting an employee session for this person (EME101)
	←	employeeSessionStartAck OK – thanks 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)
Employee inserts a note, plays a game, cashes out, etc.		
G2S_EME102	→	Employee Session Updated event conveys current delta values (stored in employee log). 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)

Note: Since individual SmartEGMs do not have a User Interface in the RLT, it is not possible to initiate employee activities or enter PINs on a SmartEGM running in the RLT. If one or more Tiger verbs to support these activities would be useful, please let us know and we can work together to define those new verbs.

User Interface Modifications

- Starting just one or two EGMs** – When the RLT Engine is started, all Enabled EGMs will typically enter the OPENING state and will start sending commsOnline commands to the host to initiate G2S communications. At times, it is convenient to be able to only start only one or two EGMs to test the application, before launching the whole set of EGMs.

In this new release of the RLT, you can start individual EGMs, *without first starting the RLT Engine*. Just select one or more EGMs on the **Manage EGMs** table, right-click to bring up the Right-click menu, then select the **Start EGM(s)** option to start the selected EGMs. Doing this changes the Engine state to “Running” so to start other EGMs, you need to repeat this procedure, or stop and restart the RLT Engine (to bring all EGMs online).



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.

NOTE: An `employeeLogList` record is overwritten by the next employee session if no meter delta activity is associated with the record.

Release Summary

In this release of the RLT, 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:

- **Informed Player Events now include the Player Log** – The SmartEGM engine was not including the player transaction log as associated data in three Informed Player events (Session Start, Session Update, and Session End). This data is needed by the host for informed player calculations.
- **Time Zone Offset processing has been added to the SmartEGM** – The SmartEGM will now process and store time zone offset information sent by the host. Timestamps throughout the tool (logs, G2S messages, etc.) now reflect the most recent G2S timezone setting.
- **Occupancy Meter Support** – Moving towards G2S 3.0, we have added Occupancy Meter Support to the cabinet class in the SmartEGM.
- **Tiger Script engine will now detect Progressive Configuration problems** – a number of Progressive configuration tests are performed when a Tiger Script is loaded (see below), and errors are reported by the tool if errors are found (before the script runs).

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 for 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). 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.

SmartEGM Modifications

- **Informed Player Session Event Data was updated** –The following Informed Player Session events were not including the Player class transaction log when transaction associated data was requested by the host. This requirement was not obvious in the informedPlayer class event table, but we realized the player log was needed upon examining the individual events.

G2S_IPE112 Informed Player Session Start

G2S_IPE113 Informed Player Session Update

G2S_IPE114 Informed Player Session End

- The `cabinetStatus.egmState` attribute now correctly transitions to “G2S_transportDisabled” when the SmartEGM determines that a communication device that is required for play has disabled the EGM because the transport is down or the host is not reachable.

NOTE: The next three modifications (Time Zone Offsets, masterResetAllowed parameter, and Occupancy Meter Support) require an updated SmartEGM configuration file that contains the newly added parameters for this functionality to be present. If you need a sample file, or assistance with updating your standard SmartEGM configuration file used in the RLT, please contact us at support@radblue.com

- **Added Support for Time Zone Offsets (cabinet class)** – The SmartEGM now supports the G2S commands for setting and retrieving Time Zone offset information.
 - This functionality is not supported by the SmartEGM unless the smartegm configuration file has valid `G2S_timeZoneSupported` parameter (value=“G2S_true”) in the cabinet device.
 - If the `G2S_timeZonesSupported` parameter is true, the SmartEGM will use the value of the `timeZoneOffset` parameter as the initial value of the `cabinetStatus.timeZoneOffset` which now controls all timestamps in the G2S messages and in the tool. The `timeZoneOffset` parameter can be updated using `setOptionChange`.
 - This current `cabinetStatus.timeZoneOffset` value can be updated via the `setTimeZoneOffsets` request from the host. The SmartEGM returns a valid `timeZoneOffsetList` to demonstrate that the update has taken place, and a G2S_CBE313 event is also generated.
 - If the host sends a `getTimeZoneOffsets` request, the SmartEGM will return a `timeZoneOffsetList` response showing the current values stored in the data model.

- **Added Support for *GTK_masterResetAllowed* parameter (cabinet class)** – The SmartEGM now supports the *GTK_masterResetAllowed* optionConfig parameter, which controls the masterReset functionality in the SmartEGM. The impact of this change follows:
 - The *setMasterRestAllowed* attribute was added to the **cabinetProfile** sent by the SmartEGM.
 - `GTK_protocolOptions.GTK_protocolParams.GTK_masterResetAllowed` was added as a valid optionConfig parameter. Changing this parameter controls the masterReset functionality.
 - If the `GTK_masterResetAllowed` parameter is set to false the SmartEGM will return a **G2S_APX008** error (Command Not Supported) to a `getMasterResetStatus`, `authorizeMasterResetStatus`, or `cancelMasterResetStatus` request from the host. s received if the *GTK_masterResetAllowed* parameter is set to false
- **Added support for the *G2S_occupancy meter* (cabinet group)** – The SmartEGM now supports a first implementation of the Occupancy Meter State Diagram (Section 5.20 of G2S 3.0), the by-product of which is a new G2S_occupancy meter in the cabinet meter group that tracks the amount of time (in milliseconds) the EGM is actually occupied.
 - Accumulation starts when money is added to the credit meter of an idle EGM (**G2S_CBE501** (EGM Occupied) is generated)
 - Accumulation is suspended when the EGM has money on the credit meter, but has been idle for longer than the `occupancyTimeOut` parameter (**G2S_CBE502** (Occupancy Suspended) is generated).
 - Occupancy accumulation ends immediately when the credit meter reaches 0 (**G2S_CBE503** (EGM Not Occupied) is generated).
 - A new `g2sOC:occupancyTimeOut` attribute was added to the cabinetProfile. `G2S_occupancyMeterOptions.G2S_occupancyMeterParams.G2S_occupancyTimeOut` was added as a valid optionConfig parameter. Changing this parameter controls the Occupancy Suspended delay timer.

Tiger Scripting Modification

- **Tiger Script Loader now validates progressives against gamePlay devices** – Several customers were having problems with Tiger script that would never hit a progressive, because the progressive hit requirements were not present in the gamePlay device configuration. To help with this, the Tiger Scripting engine now validates the progressive configuration against the referenced gamePlay device to ensure the EGM is appropriately configured. If not, and error is reported and the script will not run.

Release Summary

In this release of the RLT, we made a number of enhancements to the SmartEGM engine:

- Improved the handling of gamePlay device denomination ranges so the SmartEGM now automatically creates optimized denomination ranges for a gamePlay device, if configured to do so (using a new gamePlay device attribute in the configuration file).
- We improved the processing of gat.doVerification commands, so the SmartEGM can accurately process a doVerification request that contains a list of software components that need to be verified.
- In our continued testing of the SmartEGM, we noticed that the timing of the gat class events was not quite right. That has been corrected to match the expectations of G2S.
- We improved the SmartEGM's processing of the cabinet.authorizedMasterReset command, so that the SmartEGM detects whether the host is actually authorizing the Master Reset, and now the Master Reset process actually resets all device and class meters in the G2S data model.

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.

Remote Control Modification

- In this release, the Remote Control (REST) interface to the RLT was improved so the Engine state reported through the interface is more accurate, and now matches the state shown in the User Interface.

SmartEGM Modifications

- **Improved handling of game denomination ranges** – Denomination ranges are used in G2S to provide a shorthand method of conveying long lists of denomination values that are available for a particular gamePlay device. Originally, if a denomination range was specified in the configuration file, the SmartEGM would always send that to the host when the game denominations were requested. In this release, the functionality has become much more versatile, so you can now configure whether each gamePlay device supports game denomination ranges. If denomination ranges are supported, the SmartEGM will now analyze the denominations supported by the gamePlay device and will automatically send denomination range lists.

To ensure backwards compatibility, the SmartEGM will expand any game denomination lists that happen to be in the configuration file into a list of unique game denominations for the device.

This ensures that each denomination only gets reported once (either as a single denomination or in a denomination range) even if multiple instances of a game denomination are defined in the file.

You can now define whether the SmartEGM engine reports game denomination ranges via a new `edm:create-denom-ranges` attribute (default="false") in the base element of each gamePlay device in the configuration file (see example, below). If the attribute is not present, or its value is false then game ranges are disabled and will not be sent to the host for this device. The game play command handler was modified so it will now automatically generate appropriate game ranges, when the gamePlay device is configured to report game denomination ranges. In the standard `smartegm-config-gsa.xml` file shipped with this release, gamePlay device 3 contains this new attribute (set to "true"), so this is the only gamePlay device that will report denomination ranges.

```
<edm:game-play edm:device-id="3" 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_WA00064"
  edm:release-number="RBG_QA00002" edm:vendor-name="Radical Blue Gaming"
  edm:product-name="RBG_WhackingFrantically23" edm:secondary-game-enabled="true"
  edm:serial-number="RBG_48217" edm:theme-type="IGT_poker"
  edm:create-denom-ranges="true">
```

- **Improved processing of the `cabinet.authorizeMasterReset` command** – In examining how the SmartEGM was processing an `authorizeMasterReset` command, we found (and fixed) a number of items, listed below:
 - 1) `authorizeMasterReset` is now accepted when sent by a host with GUEST authority.
 - 2) When a `masterResetStatus` was sent in response to an `authorizeMasterReset` request, it was incorrectly being sent as a REQUEST. It's now correctly sent as a RESPONSE.
 - 3) The GTK_CBE011 [Host Authorization Received] event is now only sent when the `cabinet.masterResetStatus.authorizeStatus` attribute actually changes from G2S_pending to G2S_authorized.
 - 4) If the host sends an `authorizeMasterReset` and denies the authorization (`authorizeMasterReset.authorized = "false"`), the state now properly reverts to PENDING.
 - 5) Functionality Improvements - When a master reset occurs:
 - a. communications will be closed to all hosts
 - b. `commsOnline.metersReset` is set to true for all hosts
 - c. **ALL** device meters are reset to 0
 - d. communications are then restarted to all hosts

- **Improved processing of `gat.doVerification` commands** – The SmartEGM will now send multiple component results in `gat.verificationResult` updates when the `gat.doVerification` is for multiple components (to show the current state of each of the components).

Since the `gat.verificationResultAck` from the host may contain a reference for all of the components, the SmartEGM will not update the component log entry for the component unless the component is in a **G2S_complete** or **G2S_error** state.

- **`gat` class event timing is now more accurate** – When processing a `gat.doVerification` command containing multiple components, the events generated by the SmartEGM did not match the intention of the G2S specification document. After further clarifications by the G2S committee, the SmartEGM now sends the following G2S commands and events:

G2S Command or Event	Description (if needed)
<code>gat.doVerification</code>	[request from Host]
<code>gat.verificationStatus</code>	[response from EGM]
GAE101 [Verification Queued]	<code>doVerification</code> request is queued
GAE102 [Verification Started]	[Sent before each component is verified]
<code>gat.verificationResult</code>	[Sent by the EGM when verification of this component completes]
GAE103 [Verification Complete]	
<code>gat.verificationResultAck</code>	[response from Host]
GAE105 [Verification Result Acknowledged and Passed] or GAE106 [Verification Result Acknowledged and Failed]	
GAE102 [Verification Started]	[Sent before each component is verified]
<code>gat.verificationResult</code>	[Sent by the EGM when verification of this component completes]
GAE103 [Verification Complete]	
<code>gat.verificationResultAck</code>	[response from Host]
GAE105 [Verification Result Acknowledged and Passed] or GAE106 [Verification Result Acknowledged and Failed]	
GAE102 [Verification Started]	[Sent before each component is verified]
<code>gat.verificationResult</code>	[Sent by the EGM when verification of this component completes]
GAE103 [Verification Complete]	
<code>gat.verificationResultAck</code>	[response from Host]
GAE105 [Verification Result Acknowledged and Passed] or GAE106 [Verification Result Acknowledged and Failed]	