



New Feature



Enhancement/Upgrade



Fix



10521 All tools now 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, RST 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 RST:

- **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 RST:

Error code **G2S_CMx012** was added for invalid class or devices in the **getDescriptor** command



10701 Added a new **Enable GZIP_IN_HTTP_STACK** option to the Transport tab, located under **Tools > Configure > Engine Options**. This option determines how RST will respond when the host also supports **GZIP_IN_HTTP_STACK**. Previously, RST would accept **GZIP_IN_HTTP_STACK** messages from the host but would never send them.



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 RST, we made the following improvements:

- **Upgraded to G2S 3.0** – The RST 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 RST now supports the G2S 3.0 Employee class in the default SmartEGM Configuration file (smartegm-config-gsa.xml), by including a G2S_employee device (with sample values in the standard configuration parameters). If an inserted ID is for an Employee, a new Employee Tab on the SmartEGM control becomes active so you can generate activity at the EGM.
- **Added support for the dft Class** – The RST now supports the G2S 3.0 dft class in the default SmartEGM Configuration file (smartegm-config-gsa.xml), by including a G2S_dft device (with sample values in the standard configuration parameters). When the host initiates a dft request, the SmartEGM will process the dft transfer.

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 for each of the significant modifications and improvements:

SmartEGM Modifications

- **Added support for the dft Class** – The standard SmartEGM configuration file (**smartegm-config-gsa.xml**) now 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.

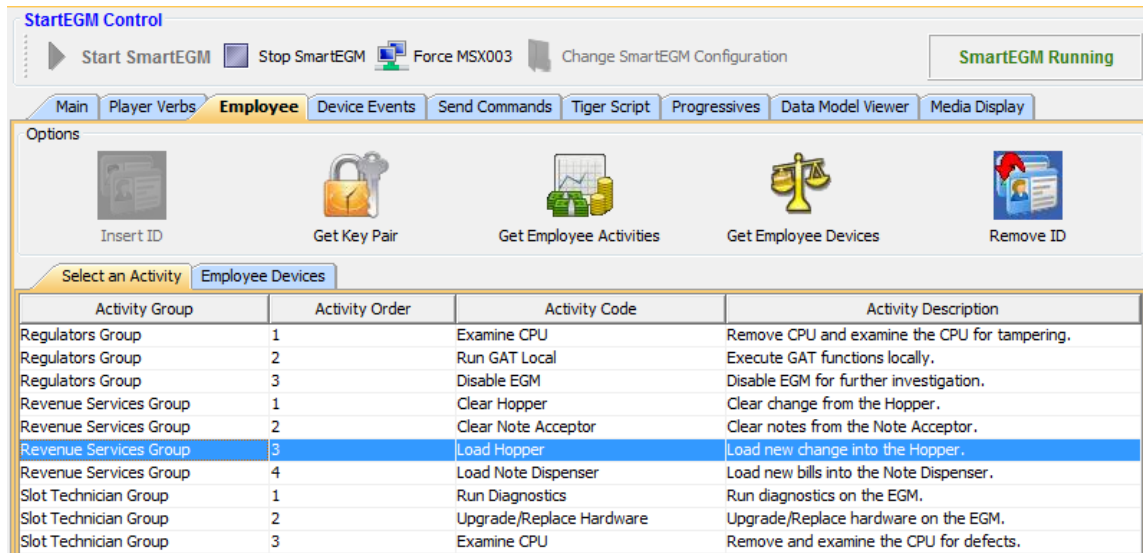
There were no RST UI changes for dft as all dft transfers are initiated via a request from the G2S host.

- Added support for the employee Class** – The standard SmartEGM configuration file (**smartegm-comfig-gsa.xml**) now 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 RST sends the following G2S requests to the host, and then activates the new Employee panel on the SmartEGM tab (see below):

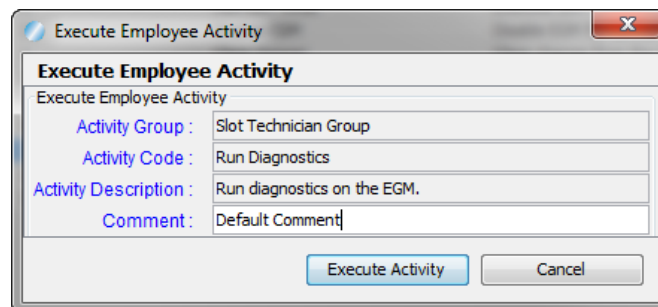
EGM Command		Host Command	Interpretive Comment (G2S_event)
Employee inserts their ID			
getIdValidation	→		This Id has just been inserted
	←	setIdValidation	It's an employee Id (idType="G2S_employee") (IDE101)
employeeSessionStart	→		Starting an employee session for this person (EME101)
	←	employeeSessionStartAck	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 recorded in the RST			
validateEmployeePIN	→		This PIN was entered
	←	validateEmployeeAck	The PIN 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 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)

User Interface Modifications

- **Employee Activity Codes** – A new Employee panel has been added to the SmartEGM Control screen:



- Once a valid employee ID is inserted, and an `employeeActivityList` has been returned by the host, the RST displays a list of Activities that are available for this employee. This list is created from the set of Employee Activities in the Employee Device Profile, filtered by the Activity Groups for which this employee is authorized (as reported in the `employeeActivityList`). Activity Groups sent by the host that are not present in the `employeeProfile` will be reported as such on the Activity list.
- To initiate an employee activity at the EGM, simply double click the appropriate row in the Activity list. If this is the first activity selected for the employee, you will be prompted for an Employee PIN, which is then sent to the host via a `validateEmployeePin` request.
- If the PIN (and keypair) are valid, you will then be able to Execute the Activity, providing a comment if desired. The activity is then sent to the host via an `employeeActivity` command.



User Interface Modifications (continued)

- **Employee Meter Activity** – Employee revenue and play activity can be easily generated using the **Player Verb** screen in the SmartEGM Control screen. As each update occurs, a G2S_EME102 (Employee Session Updated) event will be sent to the host, including the transaction log record with updated meter deltas, if desired.

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.

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.

Release Date: 03 June 2015

Release Summary

In this release of the RST, 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:

- **GZIP implementation improvements** – You can now select GZIP in HTTP Stack or GZIP Payload (as you can in the RGS). The code has been retested, a number of corrections were made. Finally, for improved diagnostics, a new column has been added to the SOAP transcript to indicate which messages are GZIP encoded.
- **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 the time zone offset information sent by the host. Timestamps throughout the tool (logs, G2S messages, etc.) now reflect the current 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 any of these tests fail (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.

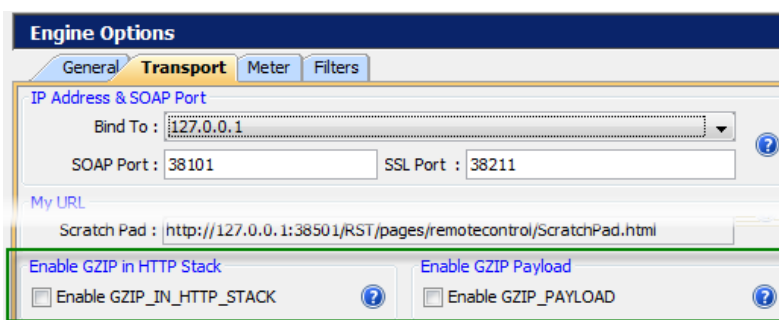
The following sections provide details for each of the significant modifications and improvements:

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.

Configuration Modifications

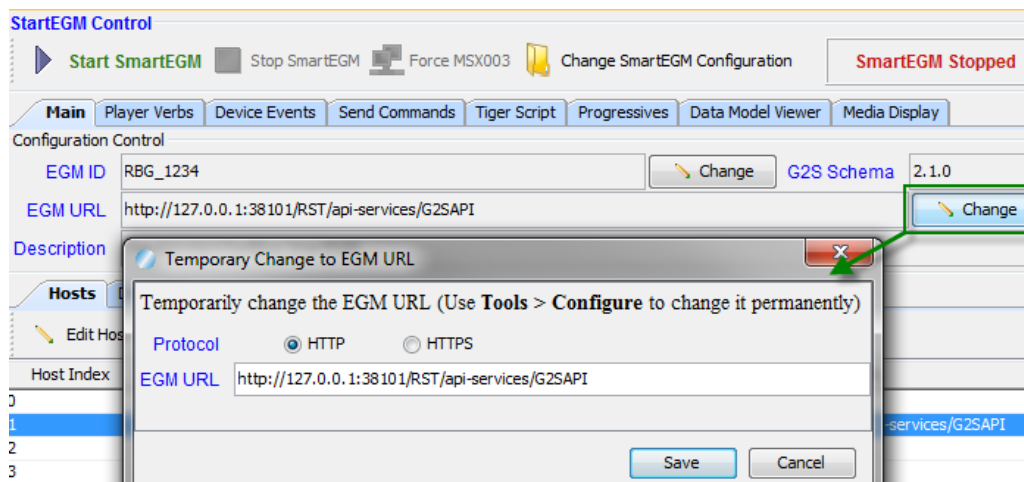
- **Changing the Transport Protocol no longer requires a restart** – If you change the transport protocol being used by the tool (between HTTP and HTTPS), the RST no longer requires a restart as you are merely switching between two active ports on the web-server. The protocol can be *permanently* changed on the Engine Option configuration screen (accessed via **Tools > Configure > Engine Options > Transport**), or *temporarily* changed via the SmartEGM layout as described in the User Interface section of these release notes.
- **Easily switch between GZIP Payload and GZIP in Stack** – The RST now supports both GZIP Payload and GZIP in Stack (the two message compression methods currently defined by GSA). To indicate that one (or both) of the compression methods are available for G2S communications, two new options have been added to the RST's Configuration Options (accessed via **Tools > Configure > Engine Options > Transport**, as shown below):



The compression method actually used in the communications is negotiated by the end-points. These configuration options merely indicate which will be supported by the RST. A comparable set of options are provided in the RGS.

User Interface Modifications

- **Game denominations are now ordered numerically** – The game denominations for the selected gamePlay device are now ordered numerically on the Play Simple Game and other game play controls on the SmartEGM's **Player Verbs** tab.
- **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**). This is a quick way of determining the compression being used on each G2S communication channel.
- **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.
- **EGM URL can be Temporarily Changed on SmartEGM layout** – The Transport Protocol used by the SmartEGM, and the URL that is reported in the `commsOnline.egmLocation` attribute can both be temporarily changed using a new button on the **Main** tab of the SmartEGM layout, as shown below. These changes are not persisted through restart, and changes made to the URL are not implemented in the RST's web-service – they are merely used for negative testing. To make changes that affect the RST's web-service, please use the Transport Configuration parameters (described above).



SmartEGM Modifications

- **Informed Player Session Event Data was updated** –The following Informed Player Session events did not include the Player class transaction log when associated transaction data was requested by the host in the event subscription. 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

- **Added Support for Time Zone Offsets (cabinet class)** – The SmartEGM now supports the G2S commands for setting and retrieving Time Zone offset information. While it will not implement changes with a *startDateTime* in the future, it will act on those that are currently in effect.
 - This functionality is not supported by the SmartEGM unless the smartegm configuration file has valid cabinet device *G2S_timeZoneSupported* parameter whose value="G2S_true".

Hint: If you are using a custom SmartEGM configuration file that you need to update, just search for "*timeZoneSupport*" in the **smartegm-config-gsa.xml** file distributed with the product to see usage examples. Need help? Send us a note: support@radblue.com
 - If the *G2S_timeZonesSupported* parameter is true, the SmartEGM will use the value of the *G2S_timeZoneOffset* parameter for the initial value of the *cabinetStatus.timeZoneOffset* which now controls all timestamps in the G2S messages and throughout the tool. The *G2S_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. A G2S_CBE313 (Time Zone Offset Changed) event is generated if the *cabinetStatus.timeZoneOffset* value changes. The SmartEGM does not support future time zone offset changes, but will implement the most recent change for which the *timeZoneChange.startDateTime* is in the past.
 - If the host sends a *getTimeZoneOffsets* request, the SmartEGM will return a *timeZoneOffsetList* response returning the values sent in the most recent *setTimeZoneOffsets* command.

- **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 *setMasterResetAllowed* 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.
- **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.
- 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.

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 requirement was not present in the gamePlay device’s configuration. To help with this, the Tiger Scripting engine now validates the progressive configuration against the referenced gamePlay device when the script loads, to ensure the EGM is appropriately configured. If not, an error is reported and the script will not run.

Release Summary

In this release of the RST, in addition to making a number of small improvements, we made more significant 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 for each of the significant modifications and improvements.

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_WAO0064"
  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]	