Release Date: 09 December 2014

Release Summary

In this release of the RLT, we have added several new methods to the REST interface, including the ability to view and change Tiger Script parameters, reset the EGM data model (for one or all EGMs), and request the RLT Version information so your client program will be able to detect what version of RLT it is communicating with.

In the SmartEGM engine, we modified our support for the keepAlive timer to correspond with rules introduced in new versions of the protocol, tightened up our validation of progressive configurations, and added a new Tiger verb that allows you to cause an EGM to hit multiple progressive jackpots during a single game cycle.

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.

SmartEGM Modifications

- **`commsDisabledAck` now updates the sync timer** – The SmartEGM engine gracefully accepted a `commsDisabledAck` from the host, and then happily ignored the *syncTimer* attribute in that message. In this new release of the SmartEGM, if a new *syncTimer* value is received, the SmartEGM uses that new value, starting a new timer for the new value. If the *syncTimer* value is less than 15 seconds, an error is logged, and the *syncTimer* is set to 15 seconds.

- **Configuration devices now use proper values for missing attributes** – commConfig, optionConfig and deviceConfig devices have no *egmEnabled*, *hostEnabled*, or *hostLocked* flags in their respective status objects. We recently learned the descriptor list for these devices MUST use the default values for the missing attributes, so the `descriptorList` now reflects that interpretation (*egmEnabled* =true, *hostEnabled* = true, *hostLocked*=false, *egmLocked*=false).

  Also, the deprecated *egmLocked* flag is now set to true only if the device ACTUALLY has the EGM locked

Tiger Scripting Modifications

- **Human.insertNote verb now includes a *count* attribute** – The **tiger:Human.insertNote** verb simulates the player inserting a note into a note acceptor device. On the RST GUI, you can specify how many notes should be inserted, so a new *count* attribute has been to the insertNote verb so you can specify the count. For backwards compatibility, the default value for this new attribute is "1". In the following example, five $10 notes are inserted into the EGM:

  ```
  <tiger:Human.insertNote tiger:count="5" tiger:denom-id="1000000" />
  ```

- **The core:pause verb now accepts a variable duration** – The **core:pause** verb is used to add fixed or random pauses as needed in a script. When creating large scripts to test systems, it became useful to be able to tune how long the script paused between each operation, via an *integerParameter* that can be changed by the tester at run-time without having to modify hundreds of pause statements in the script. In this new release of the scripting engine, you can now define an integer parameter (such as *pauseDuration*) that can then be modified as an execution parameter for the script at run-time. In the following example, a pauseDuration parameter is defined with a default value of 500 milliseconds, and then the new parameter is used in a pause statement:

  ```
  <core:parameters>
     <core:integerParameter core:name="pauseDuration" core:default-value="500" />
  </core:parameters>

  <core:pause core:duration="pauseDuration"/>
  ```

- **New method to cause a script to run only one time** – Normally, the RLT's job is to create a continuous load on a server to see how it will handle the pressure of hundreds or thousands of EGMs. When we were doing some quantitative testing with the RLT, it became apparent that it would be useful to have a method to instruct each EGM to run their Tiger script exactly one time. This way, the EGMs create a known amount of activity during the run, so the host's accounting reports can be checked to verify that they are correctly accounting for all activity from all EGMs.

  By adding the following tiger verb a script writer can instruct the RLT to not restart the script:

  ```
  <!-- Instruct RLT to not restart the script.  In RST, the script just exits. -->
  <core:exit core:message="Exit script" core:rlt-restart-script="false" />
  ```

Release Date: 26 August 2014

Release Summary

In this release of the RLT, we have added several new methods to the REST interface, including the ability to view and change Tiger Script parameters, reset the EGM data model (for one or all EGMs), and request the RLT Version information so your client program will be able to detect what version of RLT it is communicating with.

In the SmartEGM engine, we modified our support for the keepAlive timer to correspond with rules introduced in new versions of the protocol, tightened up our validation of progressive configurations, and added a new Tiger verb that allows you to cause an EGM to hit multiple progressive jackpots during a single game cycle.

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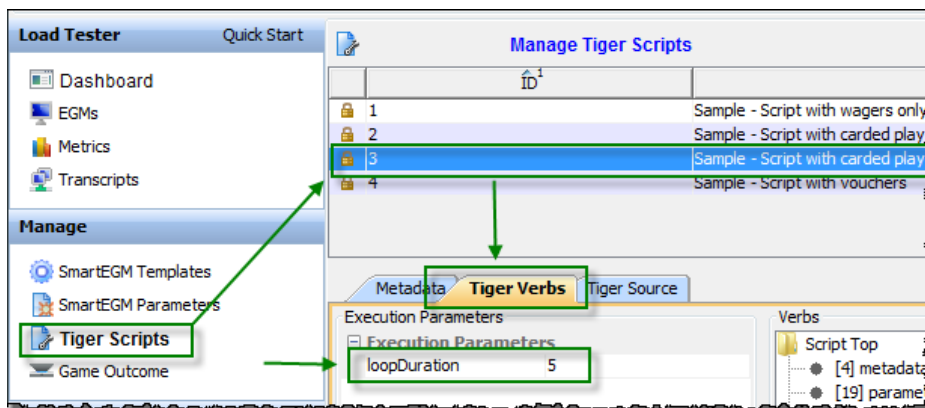
The following sections provide details on each of the significant modifications and improvements.

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 installer will download the 64 bit version of Java as part of the installation.

User Interface Modifications

- It is now possible to modify Tiger Script Execution Parameters directly in the RLT. If an active EGM is using the affected Tiger script, it will use the new value the next time the script is started. Since an EGM instance restarts its Tiger script once it has completed, this can happen automatically, or you can stop and restart the Tiger Scripts (in the EGMs' panel) to force the EGM to use the new value immediately.

SmartEGM Modifications

- **Updated keepAlive timer logic** – In the SmartEGM engine, we modified our support for the keepAlive timer to correspond with rules introduced in new versions of the protocol. Previously, the keepAlive timer was reset when any G2S message was sent to the host OR any G2S message was received from the host. Now, the keepAlive timer is only reset when messages occur on both channels (and also whenever a keepAlive is sent to the host). For more details on these new rules, please see the keepAlive command description in G2S protocol document for versions 2.x and greater.

- **Improved handpay behavior when device is hostdisabled** – The SmartEGM was properly disabling the handpay device on the `setHandpayState` command from the host, but it was not allowing any handpay functionality when the device was disabled. It now honors the `handpayProfile` "disabled" attributes that control which handpay functions are allowed when the device is disabled (for the functions that it supports).

- Notifications from the SmartEGM were being sent when the communications channel is not enabled (EGM in sync state). Now they're just discarded if they can't be sent to the host.

- Average Payback Percentage was not being calculated properly, but was always being set to the average payback percentage for the current game cycle. This has been corrected.

- There was an occasional issue where the SmartEGM would not exit option configuration mode when an `enterOptionConfigMode` command with *enable*=**false** (disable option config mode) was received, due to a race condition in the code. This has been corrected.

- The SmartEGM configuration would allow a progressive device to be configured with progressive win levels that do not exist or that don't allow progressives.

  The Smart EGM configuration progressive validation has been updated to check the following:

  1. The game play device must have *gamePlayProfile.progressiveAllowed* set to **true**

  2. The gamePlay win level being referenced must have *progressiveAllowed* set to **true**

  3. The *numberOfCredits* value in the progData table must be less than *gamePlayProfile.maxWagerCredits* for the referenced gamePlay device.

  4. The *denomId* in the progData table has to be in the list of denominations for the gamePlay device (though it does not have to be active)

  Note: If you are using a standard RadBlue configuration file, you may get an error when first starting the RLT after doing this upgrade. Just accept the error, and then change the SmartEGM configuration file to use the new version that is distributed with the tool, and all will be good. This new progressive validation caught an error in some of our old SmartEGM configuration files.

Tiger Scripting Modifications

- **Multiple Progressive Hits per Game Cycle** – The SmartEGM now supports multiple progressive hits in one game cycle via a new `playSimpleGameWithProgressiveHit` verb. This new verb allows a series of **progressive-hit-data** elements, each containing a valid *win-level-index*, which will cause multiple progressive hits in one gamePlay cycle. Each element must be valid for the selected gamePlay device. If desired, a particular **progressive-hit-data** element can be repeated multiple times. The following is an example of how the verb is used:

```
<tiger:Human.playSimpleGameWithProgressiveHit tiger:device-id="3"
    tiger:denom-id="100000" tiger:credits-to-wager-cashable="5"
    tiger:primary-win="0">
        <tiger:progressive-hit-data tiger:win-level-index="1" />
        <tiger:progressive-hit-data tiger:win-level-index="1" />
        <tiger:progressive-hit-data tiger:win-level-index="2" />
</tiger:Human.playSimpleGameWithProgressiveHit>
```

In this example, *device-id* refers to the gamePlay device being played, as do the other attributes of the `playSimpleGameWithProgressiveHit` element. Once the primary game result has been determined, three progressive hits will occur – two for win level index 1, and then one for win level index 2.

The following validations are performed when this new Tiger verb is executed:

1. The referenced game play device's profile must allow progressives.

2. Each win level referenced by the progressive-hit-data element must allow progressives

3. For each referenced win level, the engine will verify that the combo of gamePlayId, winLevelIndex, denomId, numberOfCredits MUST be an entry in the associated progressive device's progressive data table.

Remote Control Modifications

For the specific details of each of these calls, please see the RLT scratchpad. Just start the RLT, and then access the scratchpad using a browser and the following URL:
http://localhost:33501/RLT/pages/ScratchPad-RLT.html

- **Get the RLT Version (new method)** – This new method allows your program to request the version number and build date for the instance of RLT to which you are communicating. Now, you'll be able to teach your program to use newer calls based on the version number of the RLT.

- **Get Tiger Scripts (new method)** – This method provides a listing of the Tiger Scripts in the RLT database, including script ID, Description, and a list of parameter names used by the script, with current values for each.

- **Set Tiger Script Parameters (new method)** – This method allows you to change the value of an execution parameter used by a Tiger script. If an active EGM is using this script, it will use the new value the next time the script is started. Since an EGM instance restarts its Tiger script once it has completed, this can happen automatically, or you can stop and restart the Tiger Scripts to force the EGM to use the new value immediately.

- **Reset EGM Data Model (new method)** –This new method allows you to reset the data model for a selected EGM. The Engine must be stopped for this method to be successful.

- **Reset All EGM Data Models (new method)** – This new method allows you to reset the data model for all EGMs. The Engine must be stopped for this method to be successful.

Release Date: 28 MAY 2014

Release Summary

In this release, we have added quite a few usability improvements to the transcript control, including the color-coding of XML documents to make them more readable, and improving the quick filter results functionality so the transcript display remains on the selected record when you clear the filter.

More specifically to the RLT and SmartEGM (the EGM simulation engine common to the RST and RLT), we added support for the master reset extension (1e), did some work on memory management so the RLT now runs better for extended periods, and added retry logic for handpay requests.

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

- **Configured memory size parameters are now used after reinstall** – The RLT installer was not properly updating the VMOptions file, so the RLT's configured memory parameters (**Initial Memory Size** and **Max Memory Size**) were being backed-up, but were not being applied when a new version of the RLT was installed. This has been resolved.

- **RLT installer is now for a single vendor per installed instance** – This is only an issue if you are running multiple custom versions of the tool (very few are). You can now run a GSA version of the RLT plus a one company's custom extensions per installed instance. If you run into issues, we can give you two separate licenses - one for each custom extension, and then you can install two separate instances of the tool. Let Russ (russ@radblue.com) know if you think you need something like this.

- **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.

User Interface Modifications

- **Accurate Tiger Script State is now reported** – When the EGM Tiger Scripts were stopped for all EGMs, the RLT would often show the Tiger Script Status as "Stopping", but it would not advance the state to "Stopped". This has been corrected.
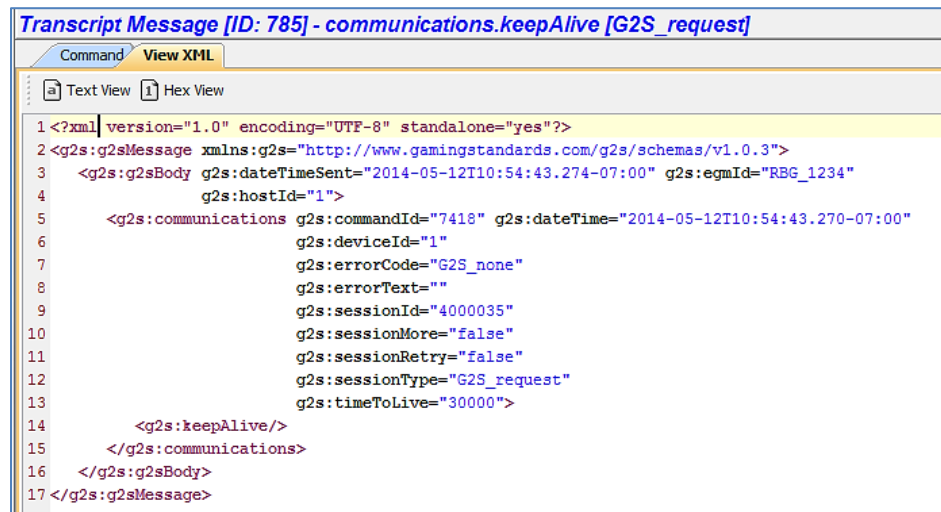
Transport and Security Modifications

- **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

- **Memory Management Improvement #1** – the JMX Monitor Service used in the RLT has been reconfigured so that it dumps the JMX (diagnostic) records every hour. The RLT now automatically keeps two days of files (gathered up when a debug.zip is created) - which will be great for debugging occasional performance issues.

- **Memory Management Improvement #2** – A memory leak was discovered when the G2S progressive class functionality was extensively exercised by the host. This has been fixed - the RLT can now run for days with a stable memory profile in all environments that we know of.
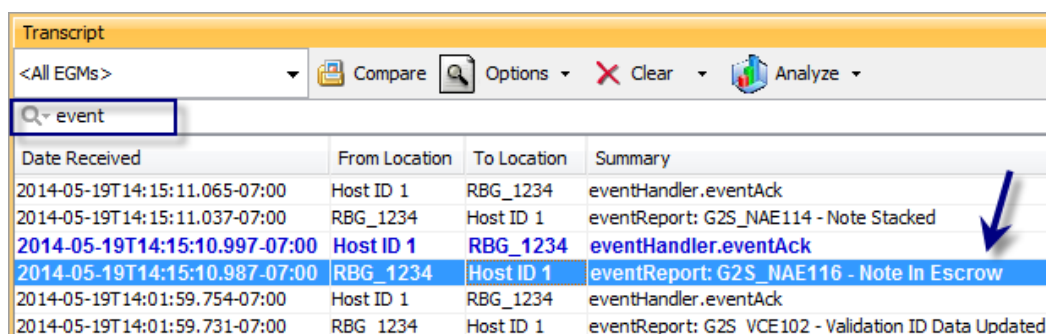
Transcript Modifications

- **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 search for a particular type of record and then select a specific message in the transcript table, that record will now always 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 no longer display in the tools.

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

Tiger Scripting Modifications

- **The `core:exit` verb was not working properly at times** – The Tiger verb `<core:exit />` was not working when it was used within a `<catch />` block inside of a `<repeat />` loop. The verb can now be used anywhere in the script to abruptly end the script if a specified condition is encountered.

SmartEGM Modifications

- **gamePlay Primary Game End event was sent at the wrong time** – The SmartEGM was sending the G2S_GPE105 event (Primary Game End) at the wrong time - The reporting of the primary game end was occurring after the processing of the secondary game was completed, but has now been moved so the event is now generated before the check for a secondary game occurs.

- **Improved gamePlay device behavior** – If an EGM's progressive device is disabled by the G2S Host, the EGM now "egm disables" the affected game play devices, instead of incorrectly "host disabling" them.

- **Missing Meter definitions -** Meter definitions were not being included with currency and game denomination meters when requested by the host.

- **Forced event subscription Improvement #1** – The SmartEGM did not handle a forced event subscription that contained event code "G2S_all ", but now it does.

- **Forced event subscription Improvement #2** – The SmartEGM was not allowing the host to clear the forced event subscription table. Now, if the host sends an empty forced event subscription table in a `setOptionChange` command, the SmartEGM will clear the table of forced events.

- **The GTECH Master Reset (gtkMR) extension (1e) has been implemented** – This cabinet class extension is used by the host to direct the EGM to return to its factory settings. Upon receiving a `cabinet.masterReset` command, the SmartEGM will now generate all expected commands and events in the master reset sequence (though it doesn't actually reset the EGM's data model to the factory state).

- **Retry logic has been added to the handpay class** – The `handpayRequest` will now be retried until the state of the handpay transaction changes to *keyedOff*. The `keyedOff` message will be retried until an acknowledgement is received. In both cases the `timeToLive` value in the handpay device's profile is used to determine the retry frequency. If the `timeToLive` value is less than 5 seconds, then 5 seconds is used.

Release Date: 28 MAY 2014

Release Summary

In this release, we have added quite a few usability improvements to the transcript control, including the color-coding of XML documents to make them more readable, and improving the quick filter results functionality so the transcript display remains on the selected record when you clear the filter.

More specifically to the RLT and SmartEGM (the EGM simulation engine common to the RST and RLT), we added support for the master reset extension (1e), did some work on memory management so the RLT now runs better for extended periods, and added retry logic for handpay requests.

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

- **Configured memory size parameters are now used after reinstall** – The RLT installer was not properly updating the VMOptions file, so the RLT's configured memory parameters (**Initial Memory Size** and **Max Memory Size**) were being backed-up, but were not being applied when a new version of the RLT was installed. This has been resolved.

- **RLT installer is now for a single vendor per installed instance** – This is only an issue if you are running multiple custom versions of the tool (very few are). You can now run a GSA version of the RLT plus a one company's custom extensions per installed instance. If you run into issues, we can give you two separate licenses - one for each custom extension, and then you can install two separate instances of the tool. Let Russ (russ@radblue.com) know if you think you need something like this.

- **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.

User Interface Modifications

- **Accurate Tiger Script State is now reported** – When the EGM Tiger Scripts were stopped for all EGMs, the RLT would often show the Tiger Script Status as "Stopping", but it would not advance the state to "Stopped". This has been corrected.

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

- **Memory Management Improvement #1** – the JMX Monitor Service used in the RLT has been reconfigured so that it dumps the JMX (diagnostic) records every hour. The RLT now automatically keeps two days of files (gathered up when a debug.zip is created) - which will be great for debugging occasional performance issues.

- **Memory Management Improvement #2** – A memory leak was discovered when the G2S progressive class functionality was extensively exercised by the host. This has been fixed - the RLT can now run for days with a stable memory profile in all environments that we know of.

Transcript Modifications

- **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:
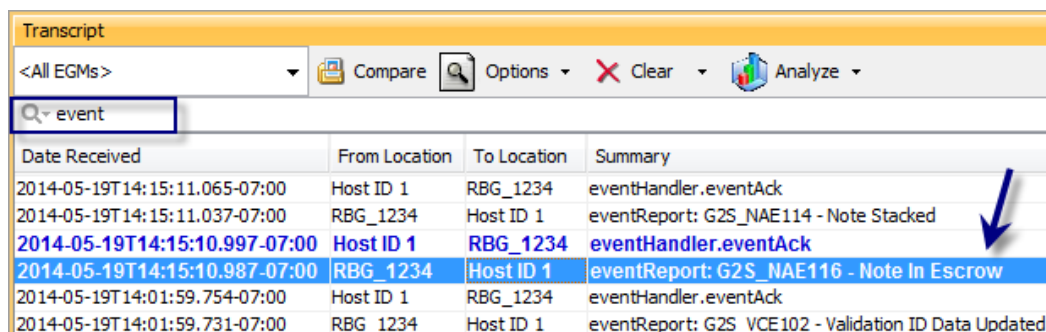

- **Improved reporting when real-time updates is disabled** – When the real-time update of the transcript is disabled, the **Realtime Update** label now goes to Red and Bold, and a message is displayed on the transcript table to make it obvious that the table is no longer being updated in real time.

- **Improved Quick Filter results display** – If you use the Quick Filter control in the transcript to search for a particular type of record and then select a specific message in the transcript table, that record will now always 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 no longer display in the tools.

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

Tiger Scripting Modifications

- **The `core:exit` verb was not working properly at times** – The Tiger verb `<core:exit />` was not working when it was used within a `<catch />` block inside of a `<repeat />` loop. The verb can now be used anywhere in the script to abruptly end the script if a specified condition is encountered.

SmartEGM Modifications

- **gamePlay Primary Game End event was sent at the wrong time** – The SmartEGM was sending the G2S_GPE105 event (Primary Game End) at the wrong time - The reporting of the primary game end was occurring after the processing of the secondary game was completed, but has now been moved so the event is now generated before the check for a secondary game occurs.

- **Improved gamePlay device behavior** – If an EGM's progressive device is disabled by the G2S Host, the EGM now "egm disables" the affected game play devices, instead of incorrectly "host disabling" them.

- **Missing Meter definitions -** Meter definitions were not being included with currency and game denomination meters when requested by the host.

- **Forced event subscription Improvement #1** – The SmartEGM did not handle a forced event subscription that contained event code "G2S_all ", but now it does.

- **Forced event subscription Improvement #2** – The SmartEGM was not allowing the host to clear the forced event subscription table. Now, if the host sends an empty forced event subscription table in a `setOptionChange` command, the SmartEGM will clear the table of forced events.

- **The GTECH Master Reset (gtkMR) extension (1e) has been implemented** – This cabinet class extension is used by the host to direct the EGM to return to its factory settings. Upon receiving a `cabinet.masterReset` command, the SmartEGM will now generate all expected commands and events in the master reset sequence (though it doesn't actually reset the EGM's data model to the factory state).

- **Retry logic has been added to the handpay class** – The `handpayRequest` will now be retried until the state of the handpay transaction changes to *keyedOff*. The `keyedOff` message will be retried until an acknowledgement is received. In both cases the `timeToLive` value in the handpay device's profile is used to determine the retry frequency. If the `timeToLive` value is less than 5 seconds, then 5 seconds is used.

Release Date: 04 MAR 2014

Release Summary

In this release, we corrected minor issues.

Engine Modifications

- An issue that caused the Remote Control user interface to freeze has been corrected.
- An issue in which RLT was incorrectly requiring OWNER permission to process a `storage.getStorageInfo` request command.
- Previously, RLT was not using the `eventHandler.eventHandlerProfile` command's *timeToLive* attribute value as the retry frequency for `eventHandler.eventReport` commands. This issue has been corrected.
- Previously, when there were multiple logs in one class (`gamePlay`, `download`, etc.), RLT used a single counter for the sequence number, resulting in skipped values in each log. RLT now uses a separate sequence number for each log.

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.