RGS Version 38 Release Notes

Release Date: 04 MAR 2014

Release Summary

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

New Features

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

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

  To create a new digest file:

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

     or

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

Copyright © 2014 Radical Blue Gaming, Inc. All rights reserved.

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

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

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

GAT Component Verification

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

To verify the components against a GAT Component Digest file:

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

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

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

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

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

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

Copyright © 2014 Radical Blue Gaming, Inc. All rights reserved.

Installation Modifications

The following configuration parameters are now retained between installations:

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

Configuration Modifications

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

Engine Modifications

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

User Interface Modifications

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

Copyright © 2014 Radical Blue Gaming, Inc. All rights reserved.

Transcript Modifications

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

Copyright © 2014 Radical Blue Gaming, Inc. All rights reserved.