RAVENBLACK
TECHNICAL SERVICES

Ravenblack Products

Multi-Agent Scheduler (RBMAS)

Administration Guide

Version: 1.3.1

Version Status: Released

Release Date: 2023/02/09

Table of Contents

1.0 Introduction

(See 9.0 Terminology for further definitions of terms used in this document)

The “Ravenblack Multi-Agent Scheduler” (**RBMAS**) provides the capability to create and manage multiple independent “agents” (each using a unique “thread”) to handle automation by scheduled WebReports. In addition to this capability, this product allows dynamic assignment of any scheduled WebReport to any one of the purpose-built agents according to the business requirements for that WebReport.

Currently without this feature implemented, the standard WebReports scheduler runs all scheduled WebReports in a serial queue, which almost always results in bottlenecks causing delays.

Using the existing WebReports functionality, any attempt to create more than one scheduling agent through server configuration, results in duplicate and conflicting executions, sometimes causing unexpected and unwanted behaviors, or even business outages.

In summary, the timeliness and accuracy of traditional WebReports scheduling is dependent on:

- How many other tasks are running.
- How often other tasks are configured to run.
- How long do other tasks take to run.

The RBMAS product provides the capability to remove most if not all these factors, allowing business sensitive tasks to run at precise intervals while large “brute force” tasks can run for as long as needed.

Besides adding the capacity for multiple non-conflicting agents, this product also provides several enhancements to existing features, as well as some performance and debugging analysis capability.

This document provides information about installing, setting up, configuration, and general usage of the RBMAS product.

2.0 Product Benefits

In this section we show some examples of how the RBMAS product improves on the existing scheduler system – and how it can fix real business issues. To start with, we use this simple (two WebReport) example for comparison between a single agent system and the RBMAS product with two agents defined.

Legacy: Simple Example of Serial Scheduling

Figure 1

This example demonstrates the “Quick Task” being delayed by the “Big Task” for about 6 minutes (including 1 minute of agent sleep time). Obviously, every additional scheduled WebReport makes the problem worse and more complicated.

Comparative Example of Ravenblack Multi-Agent Scheduler (RBMAS

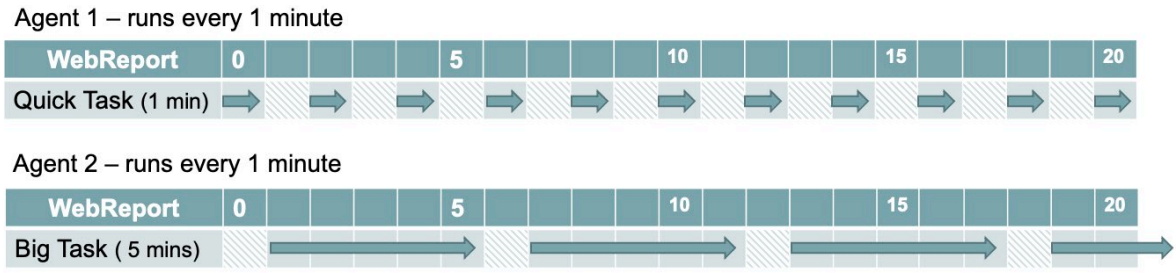


Figure 2

This diagram represents the same business scenario shown in the previous example, but with the RBMAS product used to create two separate agents that run in parallel, with each of the two WebReports assigned to a different agent. In this scenario, neither WebReport is impacted by the other and the only delay incurred is based on how long the agent sleeps (1 minute in this example) and how long each WebReport runs for, i.e. does the WebReport delay itself.

Legacy: WebReports Single Agent Scheduler - more complex example

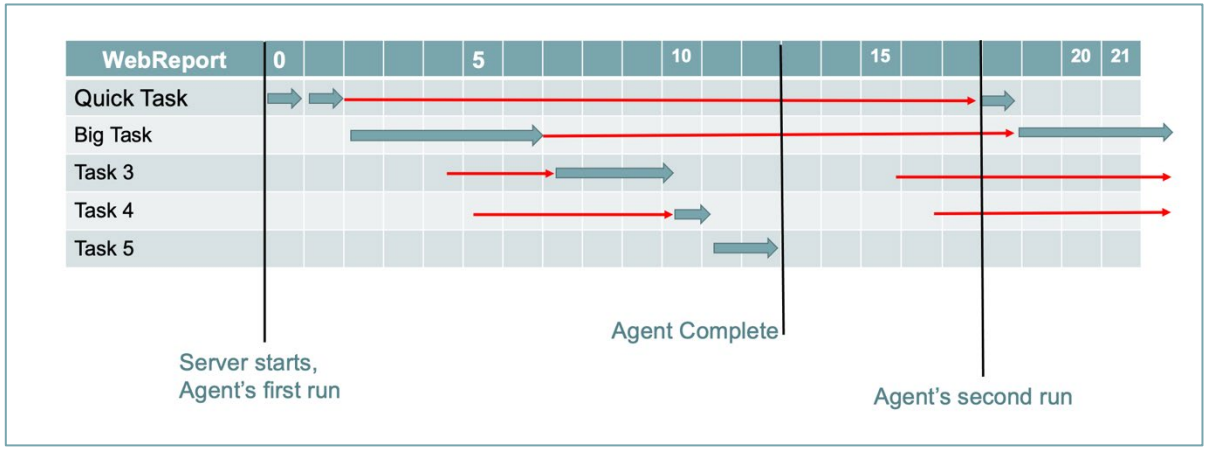


Figure 3

This diagram shows a more complex example of how scheduled WebReports typically run using the traditional, single agent approach. The red lines indicate the delay from when a WebReport was scheduled to run until the agent actually runs it. In each cycle the agent wakes up, runs all waiting WebReports (one at a time), then sleeps before starting the cycle over. The sleep time is configurable but is usually defaulted to 5 minutes as shown in this diagram. In this example, the "Quick Task" has work to do every minute but is delayed by 15 minutes due to the agent running 4 other WebReports. Some customer systems may experience delays of as long as an hour or more.

Workflow for Creating and Assigning Schedules with RBMAS

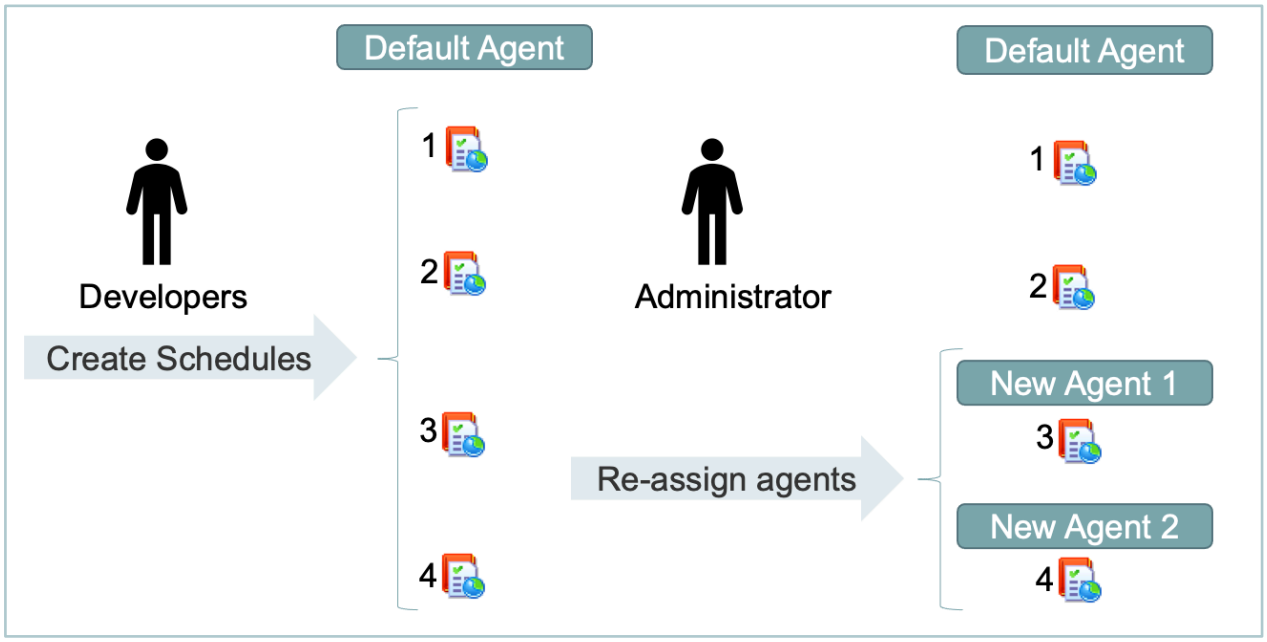


Figure 4

This diagram illustrates the basic workflow for the creation and re-assigning of WebReport schedules.

- Developer creates WebReports with destinations (that allow scheduling), then the developer selects the Schedule option for each WebReport and defines a scheduled run time (and repetition frequency).
- All WebReport schedules setup by developers are added to the workload for the “Default” agent.
- Based on business requests, performance analysis, or business complaints, the Administrator (or designated Admin resource) can then reassign (transfer) some of these scheduled WebReports to unique agents created with the RBMAS tool.
- Any WebReports that are not reassigned will remain on the default agent.
- The Developer cannot change the agent assignments unless they have access to the RBMAS **Manage WebReports Schedules** page.

3.0 Installation

This product is provided as a zip file using this naming convention: RBMAScheduler_<*version number*>.zip containing a Content Server Application (CSApp) that can be installed using the Application Management functionality on any version of Content Server since 16.0. (For earlier versions please contact support@ravenblackts.com.)

The process of installing a CSApp is documented in Content Server help, and "How to Install this (CSApp) application" document can be found in the Ravenblack Software download folder.

In general, the zip file is extracted into the ***/*csapplicationsstaging** folder and then installed using the Application Management option under the admin pages.

This product should only be installed on whichever server the existing WebReports scheduling agent is currently running on. In more recent versions of Content Server (since 21.1) this product could be installed from any server, but it should ONLY be enabled (as explained below) on the server that has the existing WebReports scheduler running on it (usually the Admin server). If the UI for this product is visible for other (non-scheduling) servers, it will be possible to re-assign WebReports, but no agent changes will be possible.

Note: in future versions it will also be possible to create agents on different servers; however, the first server setup must always be the one that has the existing scheduling agent.

On versions of Content Server after version 22.3 you will be prompted to restart the system. In prior versions of Content Server, a restart of the server should be performed. If you have the Ravenblack "Sub-tag Loader" tool, you can avoid a restart (cancel the restart for 22.3+ versions) by running the tool to build all application custom sub-tags.

Some versions of this product will be "upgradeable" meaning they can be installed without removing the previous version; however, this varies according to each release and the type of changes. Specific instructions are included in the release notes for each version.

4.0 System Enabling and Disabling

Once the RBMAS application has been installed, the existing scheduler setup is still active until such time as the RBMAS functionality is “Enabled”. There are two ways to enable the system. One of these methods is simply to access the main interface page (**Manage WebReports Schedules**) and start setting up new agents (as explained below). As soon as a new agent configuration has been saved, the new software will be active. More information on setting up new agents is available in the next section.

In addition to setting the software up by actively creating new agents, there is also an option to simply enable the RBMAS software, without changing or adding any agents.

This option also allows a “Passive mode” that is typically used to enable analysis tools to measure the performance of the existing scheduling algorithm. Both options are described in the following section.

4.1 Enabling for Passive Analysis

This method of enabling the RBMAS software is provided through a **Configuration and Maintenance** page that is described in more depth later in this document. This page can be accessed via a link on the main **Manage WebReports Schedules** page using this icon  in the top right corner.

Alternatively, the page can be accessed directly using this URL:

…`?func=csapps.launchapp&appname=RBMAScheduler&nickname=ManageSettings`

Or this one:

…`/Open/RBMASmanageSettings`

`E.g., http://somehost/otcs/cs.exe/Open/RBMASmanageSettings`

Note: This page must be accessed from the server that is handling standard WebReports scheduling. Accessing this page from another (non-scheduling) server only generates a warning screen.

4.1.1 Configuration and Maintenance page before enabling

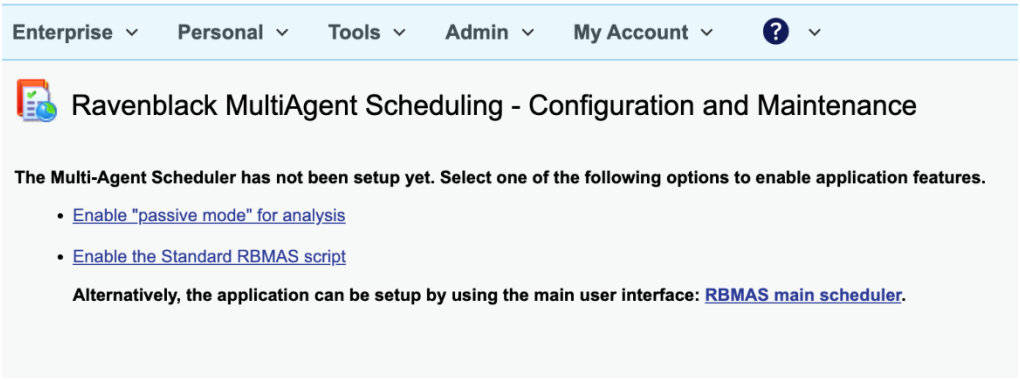


Figure 5

The screen shown above, is loaded after CSApp installation. At this point the new functionality has not yet been enabled. Clicking on **Enable the Application with Updated RBMAS script** initiates the setup of the new RBMAS software, and as a result, the single existing agent will start to run using a slightly improved RBMAS algorithm. In this mode, increased logs and performance analysis can be enabled.

Clicking on **Enable the Application in "passive mode"** also sets up the RBMAS software, but in this mode the single scheduling agent continues to run using the identical script/algorithm that is normally used by WebReports scheduling, but with the capability of generating analysis logs.

This mode is used by customers who want to analyse the current state of their system without any RBMAS functionality in order to measure the benefit of deploying multiple agents in an "Active mode".

In response to selecting either of these options, the following messages are displayed.

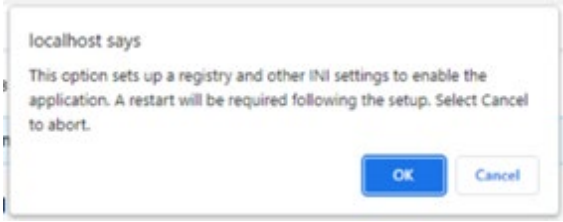


Figure 6

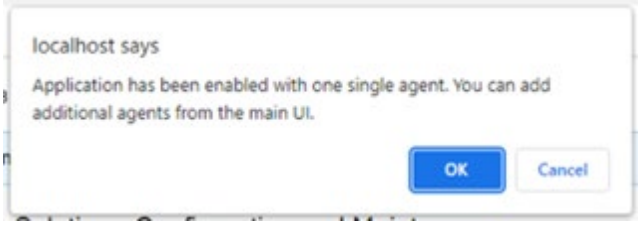


Figure 7

On completion of enabling RBMAS, the application prompts to initiate a restart. 4.3 Restarting the System shows example screenshots for the process of executing a restart. The restart can be delayed (by selecting **Continue** in *Figure 12*) but until the restart occurs, no additional analysis logs can be generated.

4.1.2 Configuration and Maintenance page after Enabling

Once the enable action and restart have completed, a full **Configuration and Maintenance** page is displayed.

Figure 8 is an example of the top table (**Miscellaneous Settings**) in the **Configuration and Maintenance** page as it normally appears. The options listed vary according to what has been enabled or disabled. If Passive Mode was selected, this screen also shows a **Disable Passive Mode** option. In either case, once this screen is visible, the **Enable Analysis** and **Enable Logs** options can be selected to enable the analysis capability. See 7.6 Enhanced Logging and Analysis for more information.

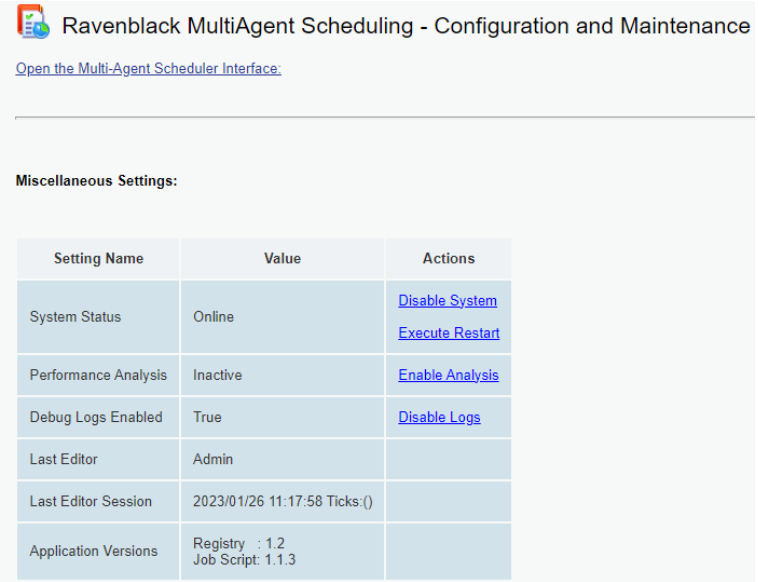
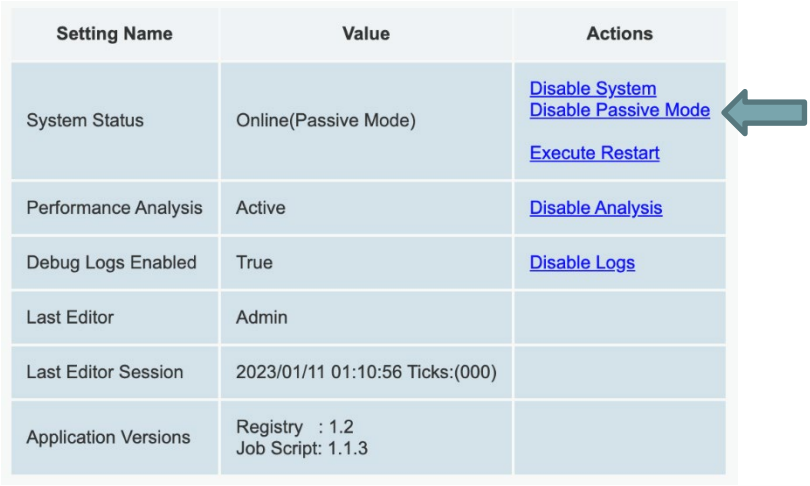


Figure 8

4.1.3 Configuration and Maintenance page after Enabling Passive Mode

Enabling the RBMAS functionality in passive mode allows the RBMAS software to measure the performance of any scheduled WebReports, while continuing to use the original scheduling algorithm (as it has existed for several years). This mode will primarily be used by customers or partners to quantify the existing performance situation without any interference (other than the measurements).

Disabling passive mode (using the option shown below) continues to run a single agent (wrschedule) but starts using a modified algorithm as it has been developed for the RBMAS product. This new algorithm may show some improvement for the single agent but is primarily optimized for running multiple agents.



Setting Name	Value	Actions
System Status	Online(Passive Mode)	Disable System Disable Passive Mode Execute Restart
Performance Analysis	Active	Disable Analysis
Debug Logs Enabled	True	Disable Logs
Last Editor	Admin	
Last Editor Session	2023/01/11 01:10:56 Ticks:(000)	
Application Versions	Registry : 1.2 Job Script: 1.1.3	

Figure 9

4.2 System Removal/Disabling

If for any reason the RBMAS software needs to be disabled, the **Disable System** option can be used (as shown in *Figure 9*). After selecting the **Disable System** option, there is a warning followed by a confirmation message:

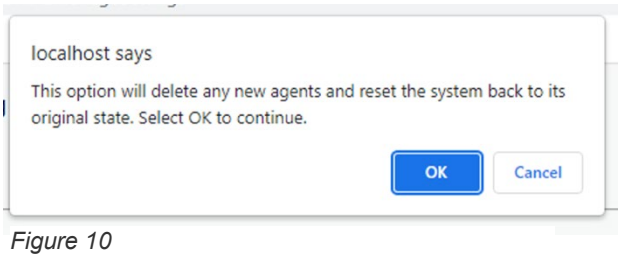


Figure 10

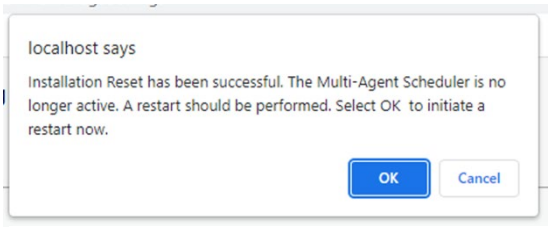


Figure 11

On completion of the **Disable System** option, restart screens will be show (identical to the screens shown in the 4.3 Restarting the System below). After the restart has completed, the screen shown in *Figure 5* is shown.

Note: This option is only available if the **Configuration and Maintenance** page was accessed from the server where scheduling has been setup.

4.3 Restarting the System

The **Configuration and Maintenance** page also has an option to **Execute Restart**. This option can be used at any time (assuming a system restart is allowed) and is useful for scenarios where a restart has been recommended but the restart wasn't executed at that time. When this option is selected, the following screen is shown:

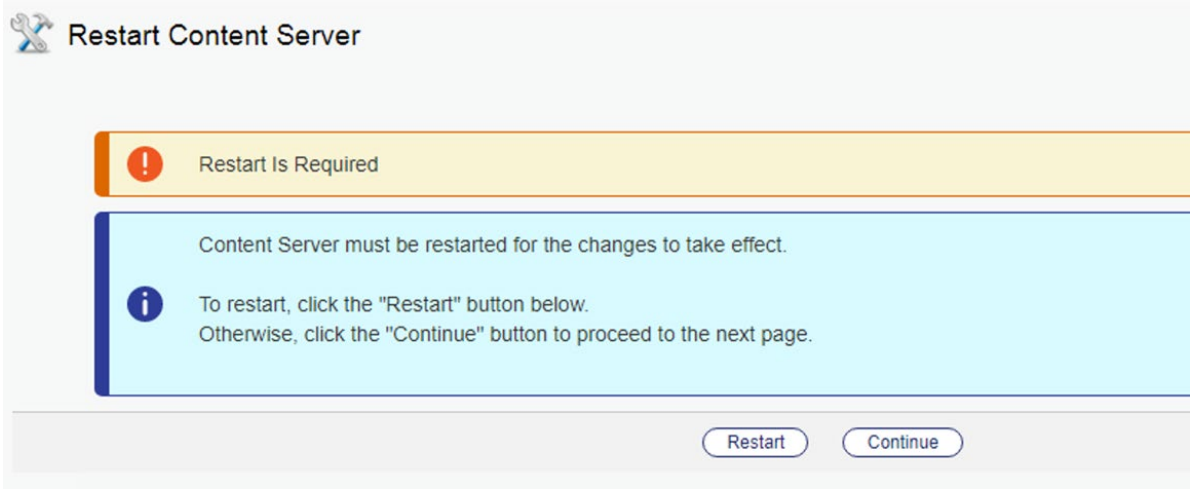


Figure 12

On completion of the restart a ‘success screen’ should appear like this:

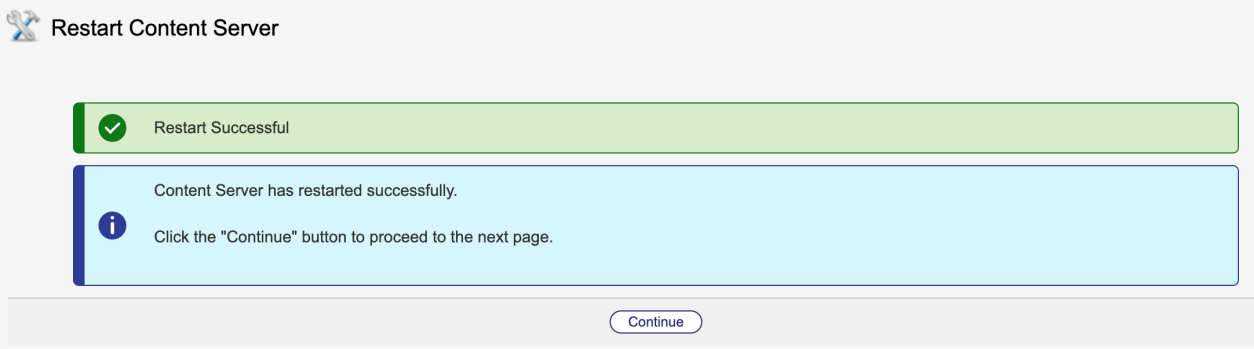


Figure 13

5.0 Managing Agents

5.1 Accessing the Manage WebReports Schedules page

Normally when the RBMAS software has been installed, an ActiveView is used to override the existing scheduling page with a new page that adds the necessary options to implement the new RBMAS features.

As a result of this override, the URL: **?func=webreports.manageSchedule** will re-direct to the new page that is explained in this section. This is normally accessed from the WebReports Administration option: **Manage WebReports Schedules** as shown here.

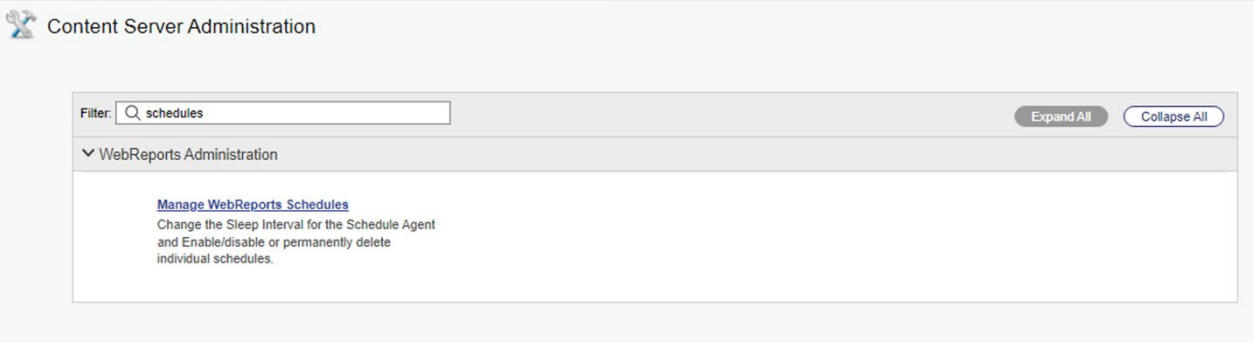


Figure 14

Note: the original (pre RBMAS) page can be accessed by adding *&avid=0* to the original URL (disabling the ActiveView).

Sometimes if there are other ActiveViews on the system with the same override type, this can interfere with the RBMAS ActiveView – causing the RBMAS UI not to display. If necessary, the Main user interface for the RBMAS software can be accessed using this URL:

```
?func=cssapps.launchapp&appName=RBMAScheduler
```

5.2 Basic Options and Layout

(See 9.0 Terminology for further definitions of terms used in this document)

5.2.1 Manage WebReports Schedules Page (for a scheduling server)

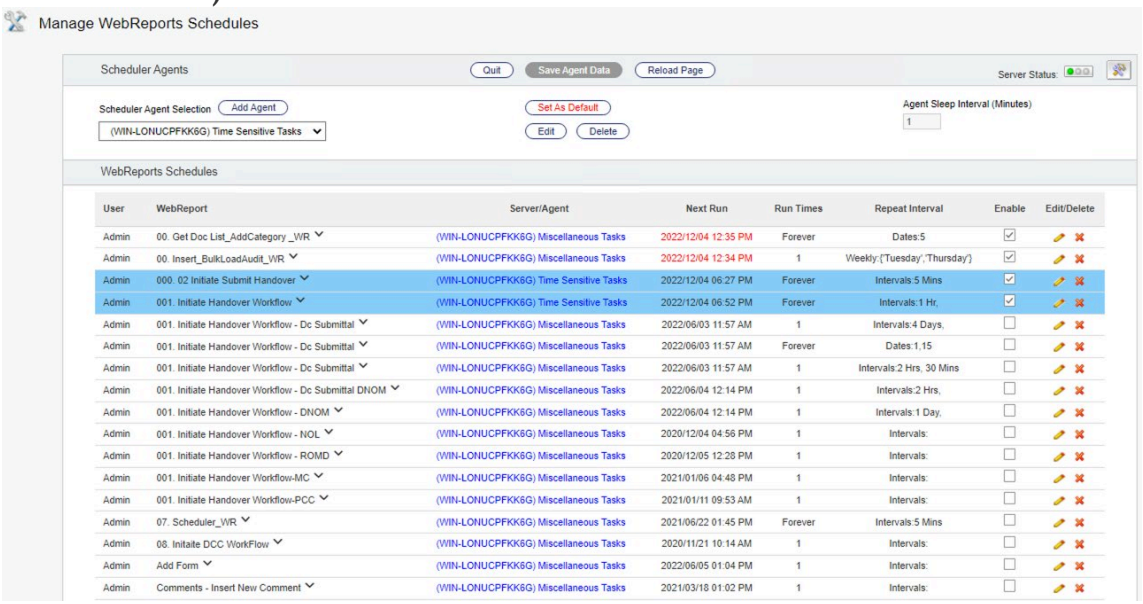


Figure 15

This is an example of the **Manage WebReports Schedules** page as it appears once the RBMAS software has been installed.

Note that there are two distinct sections (as with the legacy page), with the top section used to Manage WebReports Scheduler agents and the bottom section used to manage the WebReports that are scheduled. There is an interaction between the two sections whenever an agent is selected from the **Scheduler Agent Selection** drop-down list and any WebReports assigned to the selected agent are highlighted. These two sections are described in more depth below.

In *Figure 15*, one of the existing agents has been selected and the two WebReports that use the selected agent have been highlighted. Other WebReports can be assigned to the selected agent simply by double clicking on any line in the table. Note that any additions, deletions, or sleep interval changes made in the *Scheduler Agent Selection* section, must be committed using **Save Agent Data** before changing assignments.

The “Next Run” times that are shown in red indicate that the WebReport is late to run. This is not updated unless the page is refreshed but “hovering” over this date/time shows the lateness in real time.

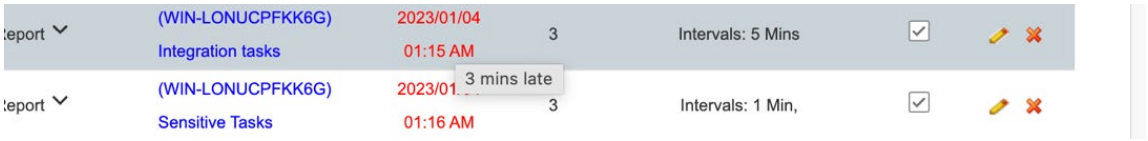


Figure 16

5.2.2 Manage WebReports Schedules Page – “Scheduler Agents”

This screenshot shows the main options for the **Scheduler Agents** section as it appears when an agent has not been selected.

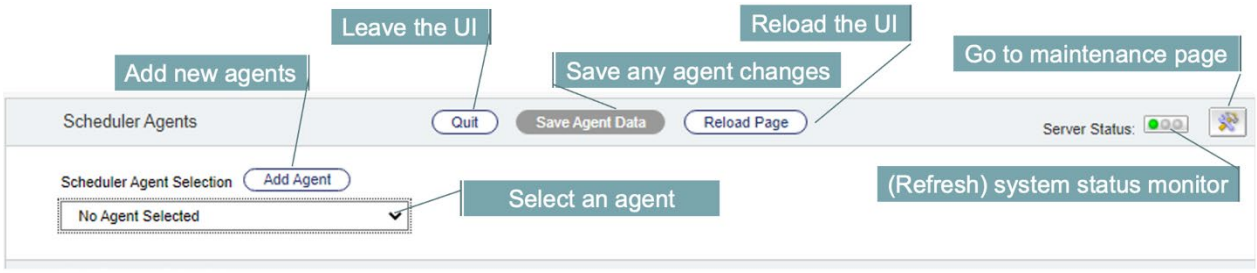


Figure 17

This screenshot shows the same section but with an agent selected from the **Scheduler Agent Selection** drop-down list. Note that when any agent is selected, a new set of options is revealed (**Set As Default**, **Edit**, **Delete**), and the agent sleep interval (how long the agent sleeps between runs) is also displayed. All these options are specifically related to the selected agent.

Figure 18

NOTE: When running this tool on macOS computers, the Default agent will not be highlighted in red, but will display a (D) after the default agent name. Additionally, when hovering over the name, Default appears to the right of the name on Windows and non-Windows computing devices. This screen shot shows a default agent name on a Mac computer.

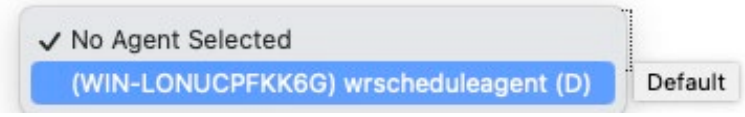


Figure 19

5.2.3 Manage WebReports Schedules Page – "WebReports Schedules"

This screenshot shows the section of the interface that is used to manage any scheduled WebReports. This section is similar to the legacy interface but with some changes noted by the attached labels. WebReports highlighted in blue are currently assigned to the selected agent. Double clicking on any non-highlighted WebReports causes them to become assigned to the selected agent. (Note that if there are any unsaved agent changes, re-assignment is not available). Similarly, double clicking on a highlighted WebReports will re-assign it to the default agent.

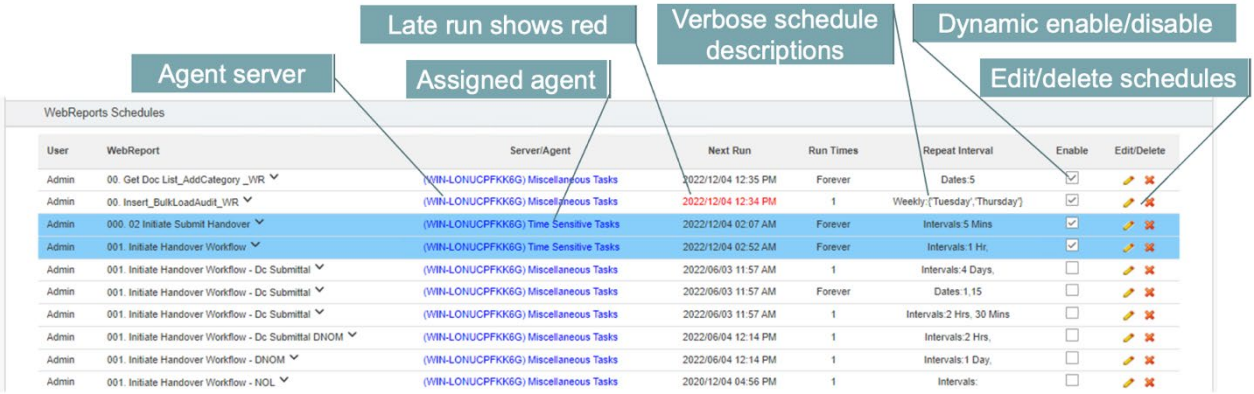


Figure 20

5.2.4 Manage WebReports Schedules Page for non-scheduling servers

This screenshot shows the screen that is shown for non-scheduling servers. No agent management is allowed from this page, but it does allow for the selection of different agents, showing which WebReports are assigned to them. From this page it is also possible to change the assignments of WebReports to agents as described previously.

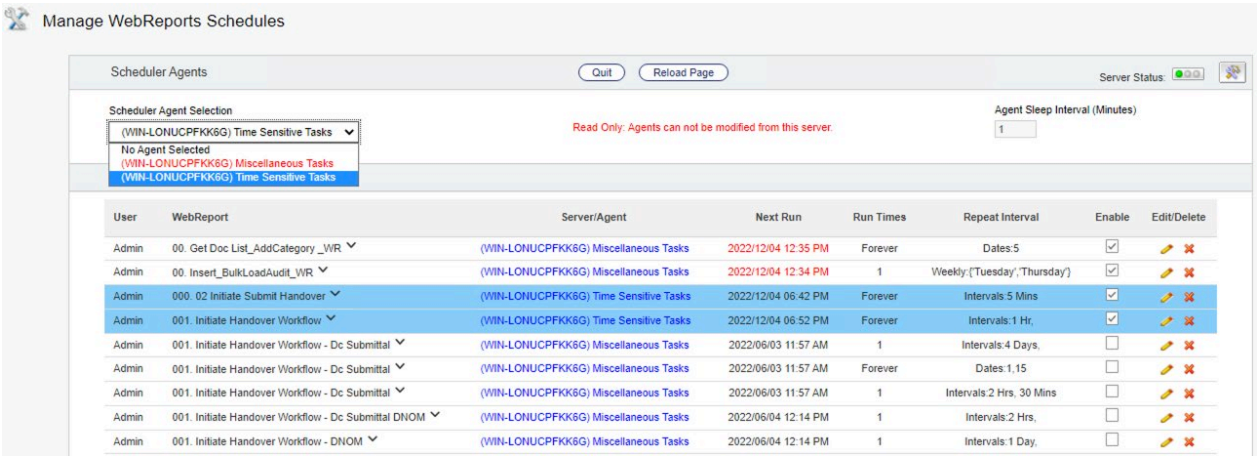


Figure 21

5.3 Adding a new Agent

The process of adding a new agent is initiated using the **Add Agent** button in the Agents Section.

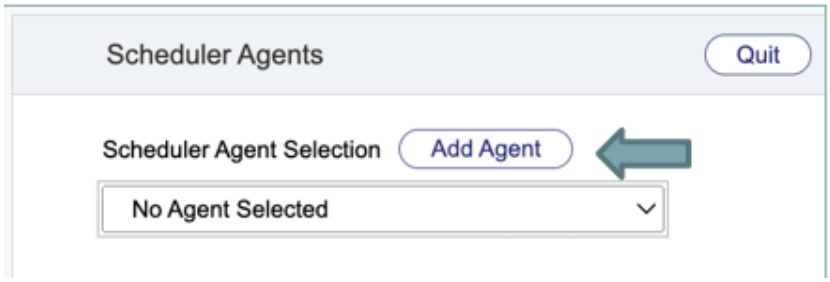


Figure 22

Once the **Add Agent** option has been selected the following screen is provided.

Figure 23

- Note that after creating a unique name, and setting a sleep interval, the Add button must be used to commit the change. Using Cancel abandons the change.
- The agent name input box provides a default but ideally a unique name should be created that describes the purpose of the agent. E.g., Business Critical Tasks, Time sensitive Tasks, Large tasks, etc.
- No changes are saved to the server until “Save Agent Data” has been selected.
- On completion of the Save action, a restart will be required.

Following any agent change, the **Save Agent Data** button is made active.

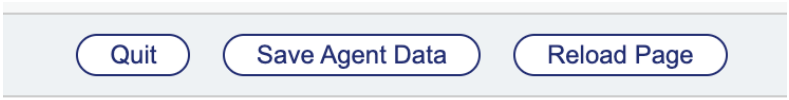


Figure 24

5.4 Editing an existing agent

The process of editing an agent is initiated by selecting an agent and then using the **Edit** button.

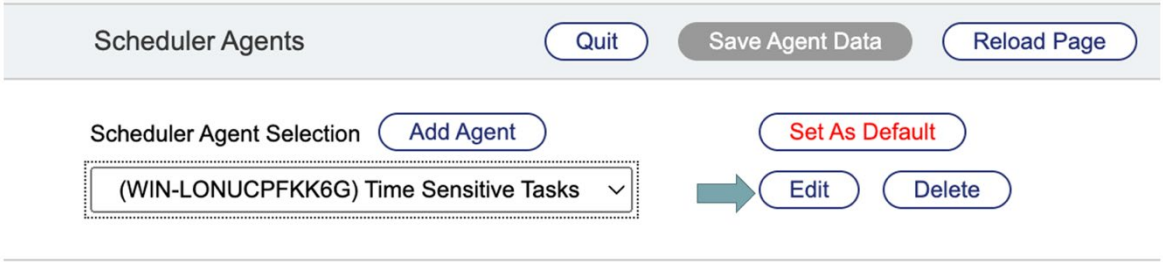


Figure 25

Once the Edit option has been selected the screen looks like this:

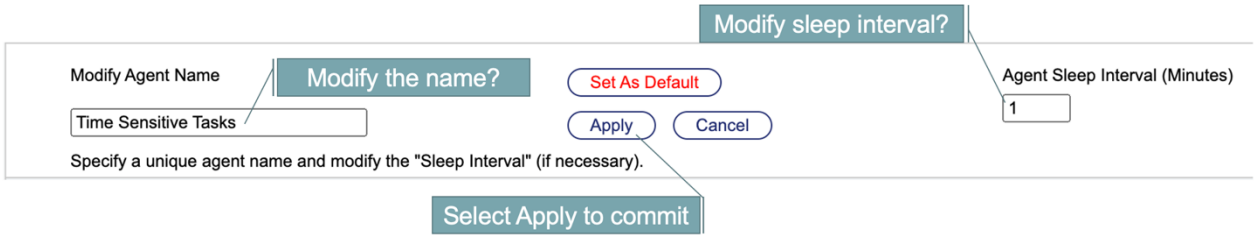


Figure 26

- Note that after making any changes, the Apply button must be used to commit the change. Using Cancel abandons the change.
- No changes are saved to the server until “Save Agent Data” has been selected.
- On completion of the Save action, a restart will be required if the Agent Sleep Interval was changed, but not if only the name was changed.

5.5 Deleting an existing agent

The process of deleting an agent is initiated by selecting an agent and then using the **Delete** button.

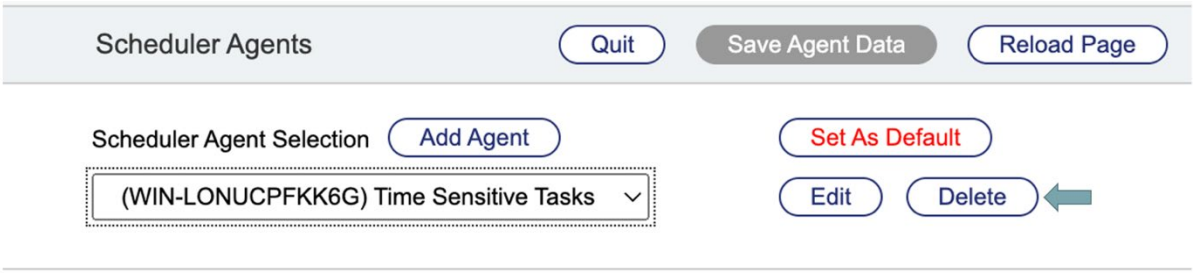


Figure 27

Once the **Delete** option has been selected the following popup may be displayed:

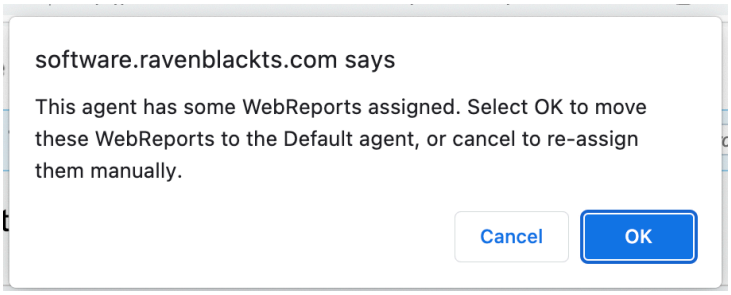


Figure 28

- As described in this message, if any WebReports are assigned to the selected agent, they can be automatically moved to the Default agent. If this is not desirable, Cancel is suggested so the agents can be moved manually if necessary.
- If the Default agent is selected, the delete action is rejected until a different agent has been set as Default.
- No changes are saved to the server until **Save Agent Data** has been selected.
- On completion of the Save action, a restart will be required.

5.6 Changing the default agent

The process of changing the default agent is initiated by selecting an agent and then using the **Set As Default** button. This button is only visible if a non-default agent is selected.

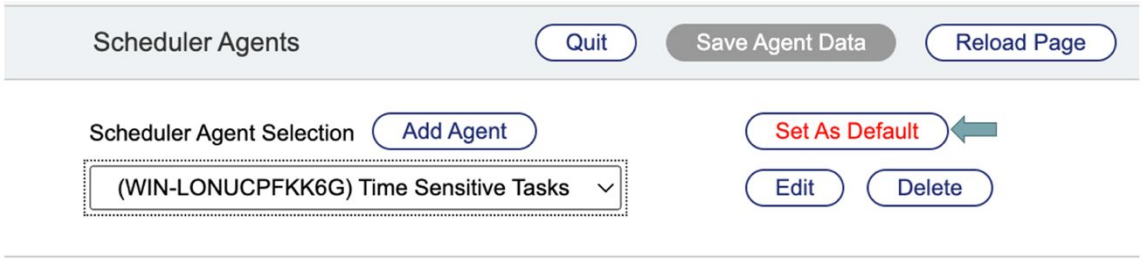


Figure 29

Once the **Set As Default** option has been selected, the following popup may be displayed:

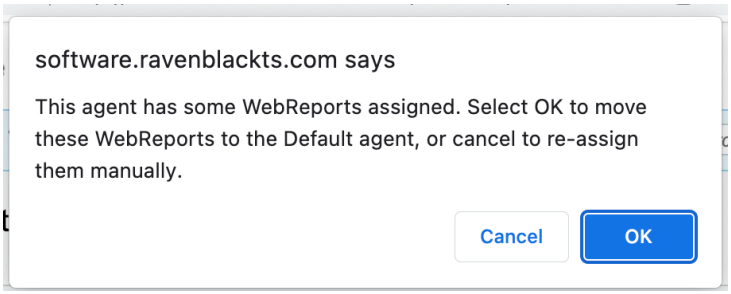


Figure 30

- As described in this message, if the agent selected to be set as “Default” has any WebReports assigned, you can select OK to move these WebReports to the agent previously set as default (essentially swapping assigned WebReports with the current “Default”). Selecting Cancel only moves the existing “Default” assignments to the new “Default” agent, leaving the original “Default” agent with no assignments.
- No changes are saved to the server until “Save Agent Data” has been selected.
- On completion of the Save action, NO restart is required.

5.7 Executing Restarts

After the **Save Agent Data** action is performed, some of these actions will require a restart. When a restart is requested, the Restart Screen is returned as shown in 4.3 Restarting the System. Once the restart is complete, the success screen shows a **Continue** button.

After selecting **Continue**, the original **Manage WebReports Schedules** page is reloaded. When the page initially loads, the status light may show a yellow light. Mousing over the status light shows this message:

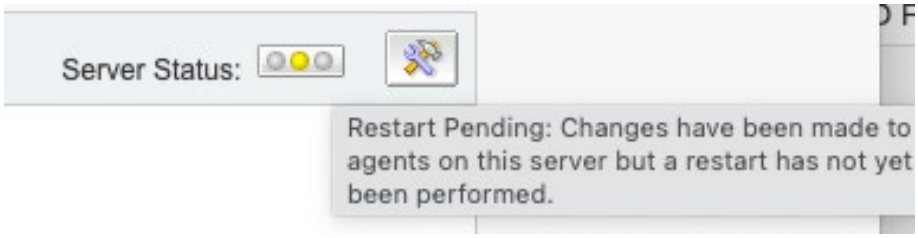


Figure 31

This message normally indicates that since making agent changes, a restart has not been performed. Additionally, there is a period following a restart when the scheduling agents have not fully started up and this status will remain yellow until the status has been changed by at least one agent.

Normally the agents should be fully started within 30-60 seconds.

The status light does not automatically update but clicking on the status light initiates a status update and the light should go green once all of the agents have successfully restarted. If the light shows red, there is a problem with one or more of the agents. There is more information about the status light in 7.1 Traffic Light Status.

6.0 Managing WebReports Schedules

6.1 Assigning WebReports to Agents

To assign a WebReport to any one of the defined agents, only two steps are required:

1. Select the target agent to be “assigned to” from the **Scheduler Agent Selection** drop-down. Note: any WebReports that are already assigned to the agent will be highlighted as shown in *Figure 15*.
2. Double-click on any row in the table (each row represents a WebReport). A progress message “Saving Agent data” is shown, followed by an “Agent information and assignments successfully saved” message (if successful). If the agent is not the “Default”, the double-click action can also be used to unassign WebReports from the selected agent, causing them to be moved back to the “Default” agent.

Notes:

- If any agent changes have been made (but not saved), assigning a WebReport is disallowed and results in a warning message.
- No restart is required after assigning/re-assigning WebReports.

6.2 Editing a Schedule

Each WebReport row includes an Edit option that allows editing of the schedule. This editing is convenient as it doesn’t require access to the WebReport via the function menu, and additionally, it allows more precise schedule settings based on the actual sleep interval of the assigned agent. The WebReports scheduling interface bases the interval precision on the sleep interval for the “Default” agent that is typically 5 minutes.

Editing can be initiated via the edit icon shown here:

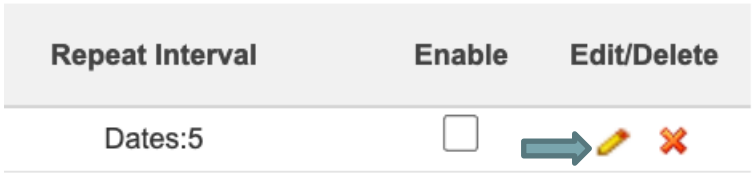


Figure 32

Selecting the Edit option provides a screen like this:

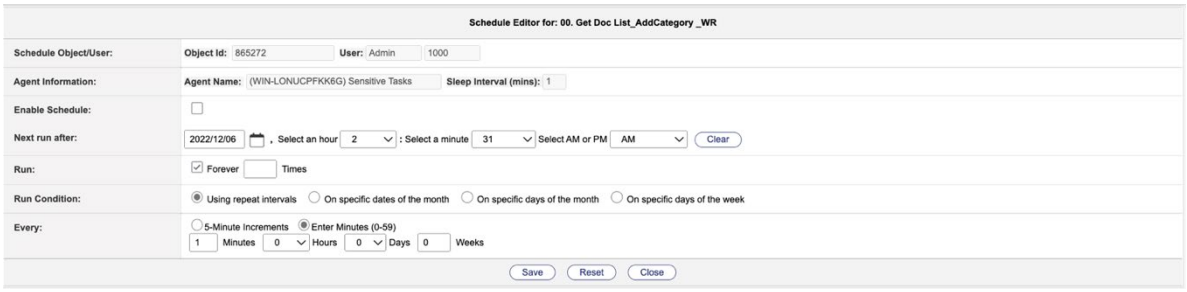


Figure 33

Except for the **Schedule Object/User** and **Agent Information** fields, this screen is essentially the same as the normal WebReports Destination->Schedule screen.

6.3 Deleting a Schedule

Each WebReport row includes a **Delete** option (see red X icon below) that allows the deletion of the schedule. Selecting this option performs the action immediately, pending a warning message.

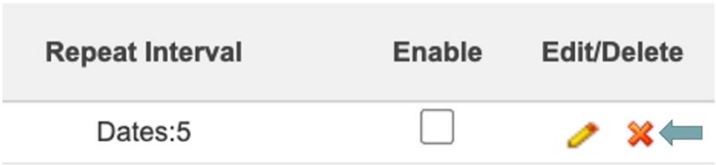


Figure 34

Selecting the **Delete** option results in a warning dialog:

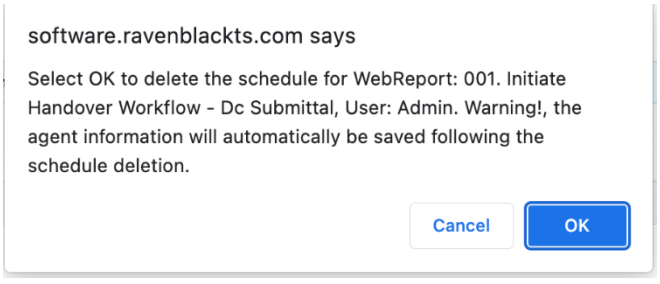


Figure 35

Notes:

- The WebReport is not being deleted by this option, only the schedule as defined in the WebReports schedule table. This does cause the item to disappear from the list of scheduled WebReports. WebReports are added to this list simply by defining a Destination that is eligible for scheduling (a Set Schedule option is visible). Re-adding to this list can be achieved by simply updating the Destination.
- As stated in this warning, an automatic saving of the WebReport/Agent assignments occurs following the deletion; however, this does not cause a restart request.

6.4 Enabling/Disabling Schedules

Any schedule can be dynamically enabled or disabled (turned on or off) using the Enable check box shown in this screenshot.

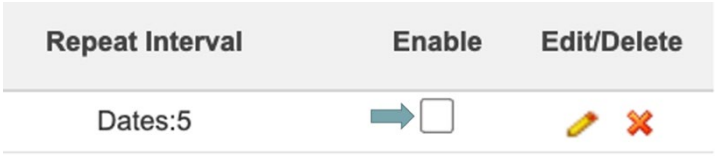


Figure 36

Following the selection (or de-selection) of this checkbox, a (brief) progress message is displayed, indicating that the schedule has changed.

7.0 Support Considerations

7.1 Traffic Light Status

The **Manage WebReports Schedules** screen includes a small “traffic light” status indicator. This light provides a quick indication of the state of the system and holding the pointer over the traffic light icon provides a status message. The traffic light is also “active” so clicking on it forces a refresh of the system status.

These are the currently supported status values and the messages that may be associated with them:

Green Status

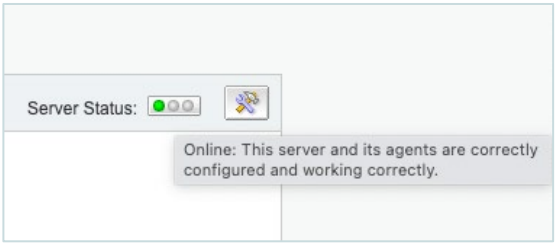


Figure 37

Other Green Status Messages

- This is the only green status message currently provided.

Yellow Status

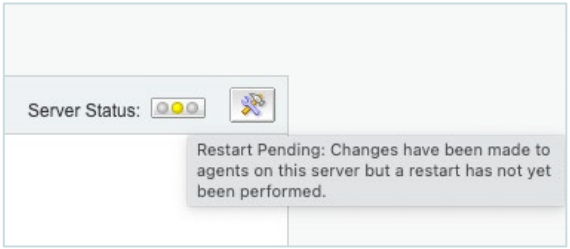


Figure 38

Other Yellow Status Messages

- “A new version of RBMAScheduler has been installed. A restart is required to fully implement any script changes”.
- “A request to the system has returned an error.”

- "A "Save Agent Data" request has detected another edit session in progress. Either Reload the page or execute "Save Agent Data" again."

Red Status

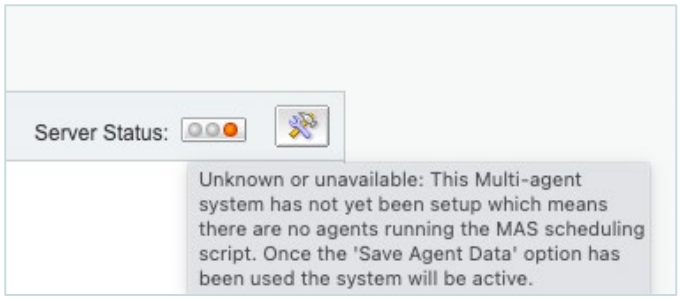


Figure 39

This message is normally seen at any time when the RBMAS Manage WebReports Schedules page is viewed, and the functionality has not been setup. It could also be seen in the unlikely event that the RBMAS "registry" has become corrupt.

Other Red Status Messages

- This is the only Red status message currently provided.

7.2 Configuration and Maintenance Page

The **Configuration and Maintenance** page provides a variety of useful options and data fields that can be useful for debugging or support functions. Some features are designed for support of the RBMAS tool and some features can be used to support WebReports that are being run by the scheduler.

This page is divided up into four sections that are explained below.

The first section of this page was briefly covered (with options related to enabling and disabling the system) in 4.0 System Enabling and Disabling.

The Configuration and Maintenance page can be accessed from the main Manage WebReports Schedules page using this icon  in the top right corner.

Alternatively, the page can be accessed directly using this URL:

…**`?func=csapps.launchapp&appname=RBMAScheduler&nickname=ManageSettings`**

Or this one:

…**`/Open/RBMASmanageSettings`**

Section 1 – Miscellaneous settings: normal, enabled, system

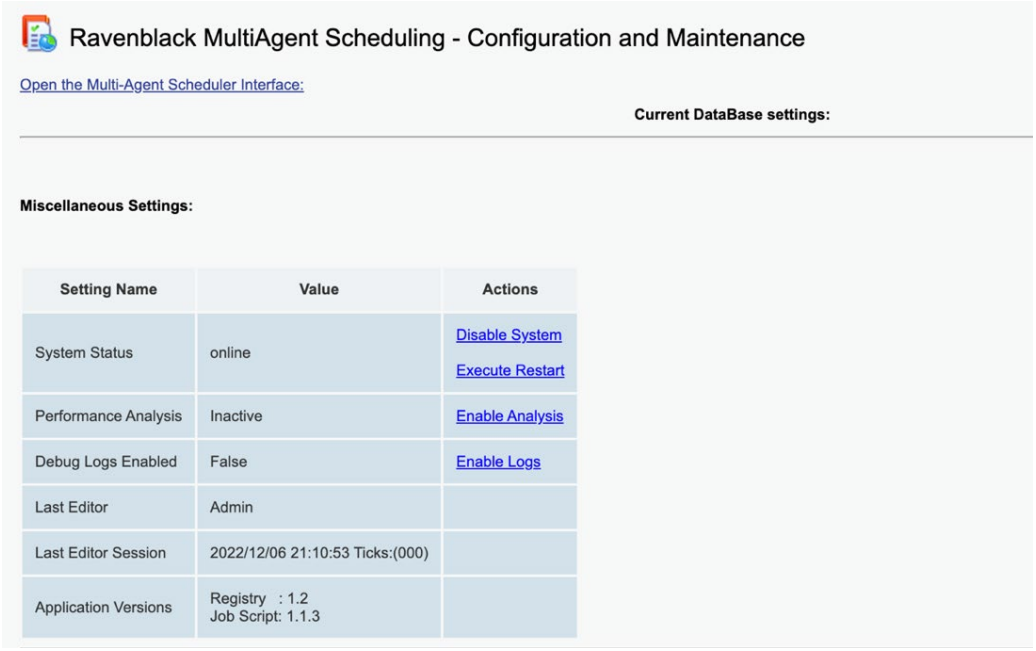


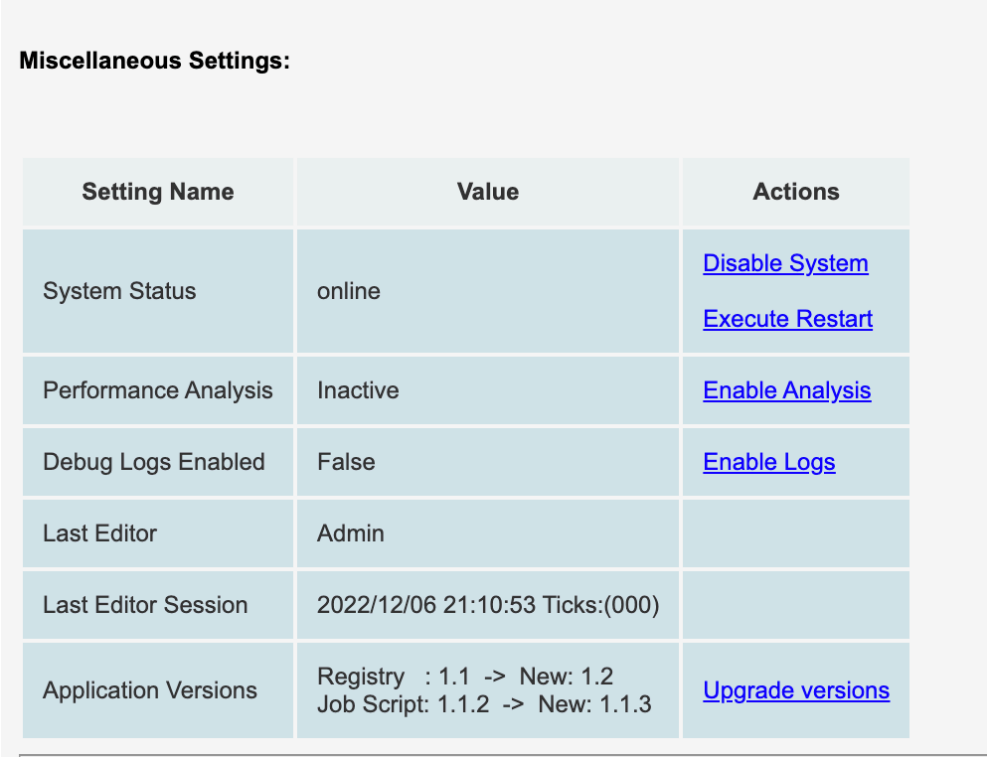
Figure 40

This is the typical layout for this section when the system has been enabled.

- The System Status is “online”.
- For an online system, the options: **Disable System** and **Execute Restart** are visible whereas for an offline system a simple page with **Enable** options is provided. These options are described in more detail in 4.0 System Enabling and Disabling.
- **Enable Analysis** and **Enable Logs** are covered under 7.6 Enhanced Logging and Analysis.
- **Last Editor** and **Last Editor Session** reflect who has most recently established a session with the **Manage WebReports Schedules** page and when they established it. These fields can normally be ignored unless Ravenblack support personnel request this information.

Application Versions reflects the current version of the “Job Script” and the “Registry”. These two components work together to manage the scheduling algorithm. Normally these two versions are simply displayed, but if a new version of RBMAS has been installed, any version upgrades are specified as shown in *Figure 41*.

Section 1 - Miscellaneous Settings: Upgrade required after install



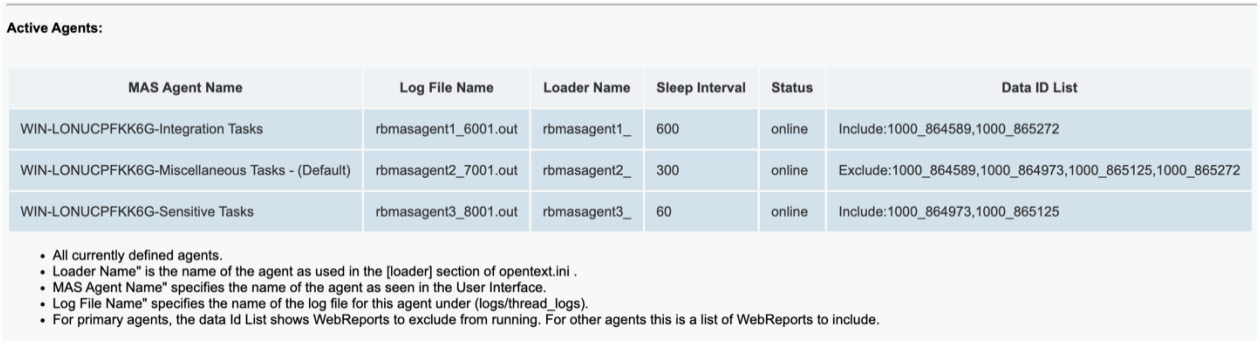
Miscellaneous Settings:

Setting Name	Value	Actions
System Status	online	Disable System Execute Restart
Performance Analysis	Inactive	Enable Analysis
Debug Logs Enabled	False	Enable Logs
Last Editor	Admin	
Last Editor Session	2022/12/06 21:10:53 Ticks:(000)	
Application Versions	Registry : 1.1 -> New: 1.2 Job Script: 1.1.2 -> New: 1.1.3	Upgrade versions

Figure 41

- In this screenshot, **Application Versions** shows the current versions and the new versions for the Registry and Job Script.
- When the system is in this state, the Actions column includes an option to: **Upgrade versions**.
- Selecting **Upgrade versions** initiates an upgrade as described in 7.5 Upgrade Process.

Section 2 – Active Agents



Active Agents:

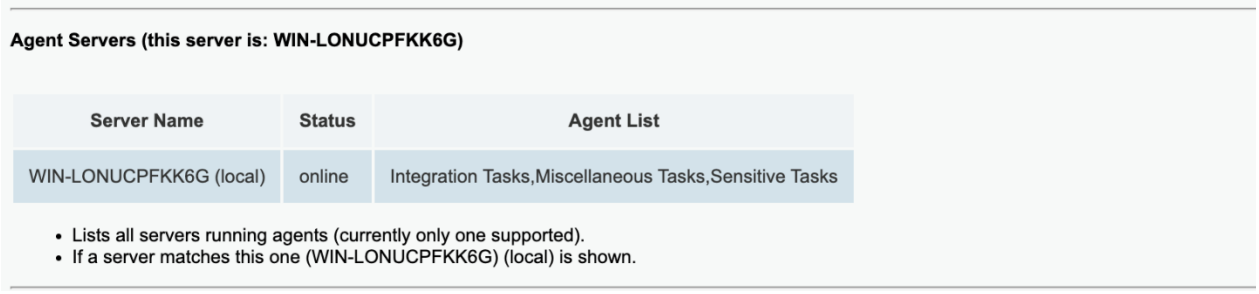
MAS Agent Name	Log File Name	Loader Name	Sleep Interval	Status	Data ID List
WIN-LONUCPFKK6G-Integration Tasks	rbmasagent1_6001.out	rbmasagent1_	600	online	Include:1000_864589,1000_865272
WIN-LONUCPFKK6G-Miscellaneous Tasks - (Default)	rbmasagent2_7001.out	rbmasagent2_	300	online	Exclude:1000_864589,1000_864973,1000_865125,1000_865272
WIN-LONUCPFKK6G-Sensitive Tasks	rbmasagent3_8001.out	rbmasagent3_	60	online	Include:1000_864973,1000_865125

- All currently defined agents.
- Loader Name" is the name of the agent as used in the [loader] section of opentext.ini .
- MAS Agent Name" specifies the name of the agent as seen in the User Interface.
- Log File Name" specifies the name of the log file for this agent under (logs/thread_logs).
- For primary agents, the data Id List shows WebReports to exclude from running. For other agents this is a list of WebReports to include.

Figure 42

- This example shows a system with three agents.
- *Agent Name* column shows the Server that houses the agent, and the full name as is seen in the **Manage WebReports Schedules** interface.
- *Log File Name* is the exact filename as it is found in the /logs/thread_logs folder (on older systems the files may simply be under /logs/.
- *Loader Name* shows the name of each agent as it is specified in the opentext.ini file. Each one of these agent names is found under **[loader]**, and each one has its own defined scheduling section, e.g. **[rbmasagent1_]**.
- *Sleep Interval* represents how long each agent sleeps (in seconds) between runs.
- *Status* reflects that the agent is operating normally (or not).
- *Data ID List* for the “Default” agent shows which schedules it does not run, whereas the *Data ID List for* the non-Default agents show which schedules they run.
- There are several notes below this data that should help to explain these columns.

Section 3 – Agent Servers



Agent Servers (this server is: WIN-LONUCPFKK6G)

Server Name	Status	Agent List
WIN-LONUCPFKK6G (local)	online	Integration Tasks,Miscellaneous Tasks,Sensitive Tasks

- Lists all servers running agents (currently only one supported).
- If a server matches this one (WIN-LONUCPFKK6G) (local) is shown.

Figure 43

- Currently only one server is supported but in future this will show each server and which agents it is managing.
- Each server has its own status.

Section 4 - Raw Schema

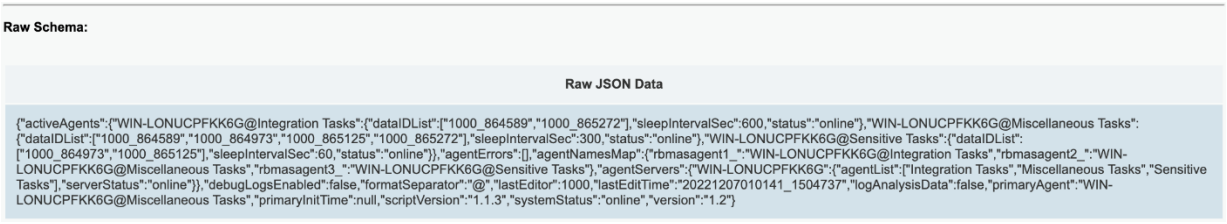


Figure 44

- This section is not normally useful for customer personnel. Ravenblack support may request that this block of data is copied and shared when required for debugging purposes.

7.3 Useful Request Fields

When a WebReport is run by the scheduler, there are parameters that are added to the request to run the WebReport. These fields can be accessed using the standard parameters tag, i.e.
`[LL_REPTAG_&<parmname> /]`

The fields that are currently available are listed in this table:

Field Name	Example	Description
agentName	`[LL_REPTAG_&agentName /]`	The name of the agent that is running the WebReport.
IamPrimary	`[LL_REPTAG_&IamDefault /]`	Returns true if the agent running the WebReport is the default agent.

7.4 Interpreting Logfiles

Like other Content Server threads, all output from scheduled agents is also stored in unique log files. Because there is no interaction with users, the only way to monitor the progress of these agents (besides information provided by the RBMAS interface) is to review these log files.

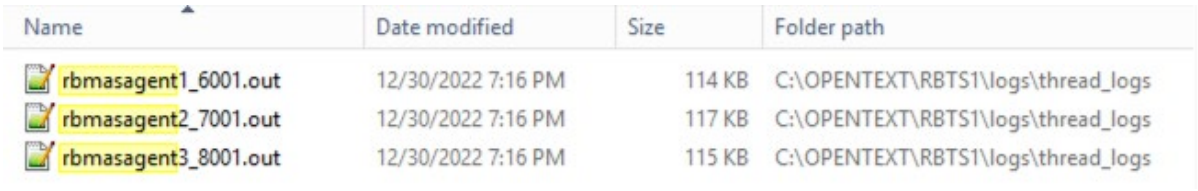
Much of the output from each scheduler agent is similar or identical to other log files; however, the RBMAS agents have been designed to enhance the traditional output, and optionally to enable an enhanced logging feature as explained in 7.6 Enhanced Logging and Analysis.

This document does not provide an exhaustive explanation of these logs, but some common conventions and themes are explained here. For further information on debugging issues through log files, please contact support@ravenblackts.com.

7.4.1 Log File Basics

Agent log files can be found in the standard location for thread logs under the standard log file folder. This is usually: **.\logs\thread_logs**

Within this folder there are several different thread log files, but any threads associated with agents will have a 4-5 digit number appended to the file name within the format: nn001 where nn is a 1-2 digit number. Typically, RBMAS log files will look like this:



Name	Date modified	Size	Folder path
rbmasagent1_6001.out	12/30/2022 7:16 PM	114 KB	C:\OPENTEXT\RBTS1\logs\thread_logs
rbmasagent2_7001.out	12/30/2022 7:16 PM	117 KB	C:\OPENTEXT\RBTS1\logs\thread_logs
rbmasagent3_8001.out	12/30/2022 7:16 PM	115 KB	C:\OPENTEXT\RBTS1\logs\thread_logs

Figure 45

In Section 2 – Active Agents
the naming convention is explained and mapped to specific agents. For example, in this excerpt from the ActiveAgents table on a system with RBMAS, we can see that the log file for the agent named: "**Adhoc WebReports**" (on server WIN-LONUCPFKK6G) is: **rbmasagent1_6001.out**

MAS Agent Name	Log File Name	Loader Name
WIN-LONUCPFKK6G-Adhoc WebReports - (Default)	rbmasagent1_6001.out	rbmasagent1_
WIN-LONUCPFKK6G-Integration Tasks	rbmasagent2_7001.out	rbmasagent2_
WIN-LONUCPFKK6G-Sensitive Tasks	rbmasagent3_8001.out	rbmasagent3_

Figure 46

7.4.2 Standard Log File Conventions

When reviewing log files, it is useful to know that the first part of any thread log file shows the progress of the initial loading of the agent. An agent thread log file should look like any other thread log file. The end of the start-up script and beginning of normal logs is marked by the following lines:

… **`INFO Startup script finished.`**

… **`12/30/2022 19:02:31 INFO [core] OScript thread startup finished.`**

Each time that the agent wakes up to do some more work, the logs for that run are bracketed by the Starting Script and Done with Script logs.

12/30/2022 18:22:07 INFO [3343556136616] Starting script

. . . *RBMAS Script output is here*

12/30/2022 18:22:07 INFO [3343556147022] Done with Script

Within these standard log markers, RBMAS generates unique log information in addition to logs that were generated by the original WebReports Scheduler. The RBMAS logs are marked with unique characters that are described in 7.4.3 RBMAS Conventions.

Note that for each WebReport that is run, any logs generated by that WebReport appear within the RBMAS logs, but each run will also generate a log indicating which WebReport (and user context) are being run as shown in this example.

```
##$ WebReport Agent: Running report "_One minute test Report" (nodeid 2043400) (userid 1000)
```

7.4.3 RBMAS Conventions

The following table shows the different prefixes used with logs generated by RBMAS. The unique characters are designed to make it easier to search for specific log sections, and/or to programmatically extract information. Note that all log types are prefixed with "##" making it possible to search for all log types if necessary.

Some of these logs are based on the original scheduler and some have been added by RBMAS.

Prefix	Log Type
##$	Progress messages, main headings (unconditional).
##@	Progress messages (unconditional).
##	Error Messages (unconditional).
##?	Conditional logs (require the Debug Logs option to be enabled).

The following two sections contrast typical logs with and without Debug Logs Enabled (*Figure 40*):

Debug Logs Enabled = false

```
12/30/2022 23:32:20 INFO  [3362169526928]   Starting script
12/30/2022 23:32:20 INFO  WebReports Agent: ##@=== Multi Agent WebReport Schedule script version 1.1.3 is running. Start time 1623944
12/30/2022 23:32:20 INFO  WebReports Agent: ##@=== rbmasagent3_ (Sensitive Tasks) - Agent running: WIN-LONUCPFKK6G@Sensitive Tasks; P
12/30/2022 23:32:20 INFO  WebReports Agent: ##$
12/30/2022 23:32:20 INFO  WebReports Agent: ##$ Subsequent run of agents
12/30/2022 23:32:20 INFO  WebReports Agent: ##$
12/30/2022 23:32:20 INFO  WebReports Agent: ##$ WebReport Agent: Running report "_One minute test Report" (nodeid 2043400) (userid 10
. . . WebReports logs . . .
```

Figure 47

Debug Logs Enabled = true

```
12/30/2022 23:44:38 INFO  [3362907432932]   Starting script
12/30/2022 23:44:38 INFO  WebReports Agent: ##@=== Multi Agent WebReport Schedule script version 1.1.3 is running. Start time 169773
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Loader Agent Name: rbmasagent3_ - Time is 16977357
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== App Agent Name: Sensitive Tasks - Time is 16977357
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Full Agent Name: WIN-LONUCPFKK6G@Sensitive Tasks - Time is 16977357
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Server Name: WIN-LONUCPFKK6G - Time is 16977357
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== agentInfo: A<1,?,'dataIDList'={'1000_2043400','1000_864589','1000_864973','1000_8
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== MAS Registry: A<1,?,'activeAgents'=A<1,?,'WIN-LONUCPFKK6G@Adhoc WebReports'=A<1,?,
12/30/2022 23:44:38 INFO  WebReports Agent: ##@=== rbmasagent3_ (Sensitive Tasks) - Agent running: WIN-LONUCPFKK6G@Sensitive Tasks; I
12/30/2022 23:44:38 INFO  WebReports Agent: ##$
12/30/2022 23:44:38 INFO  WebReports Agent: ##$ Subsequent run of agents
12/30/2022 23:44:38 INFO  WebReports Agent: ##$
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Working list of unique IDs for this agent: {'1000_2043400','1000_864589','1000_86
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== SQL command: (SELECT dtr.name, wrs.* FROM    WEBREPORTS wrs INNER JOIN DTree dtr (
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Total of 1 WebReport schedule instances have been selected from WEBREPORTS table :
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Schedule instance match found in form <USERID>_<WR NODEID>: 1000_2043400 - Time is
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== WebReport node name for matching instance: _One minute test Report - Time is 1697
12/30/2022 23:44:38 INFO  WebReports Agent: ##?=== Schedule instance 1000_2043400 is due to run. - Time is 16977359
12/30/2022 23:44:38 INFO  WebReports Agent: ##$ WebReport Agent: Running report "_One minute test Report" (nodeid 2043400) (userid 1(

. . . WebReports logs . . .

12/30/2022 23:45:23 INFO  WebReports Agent: ##?===&& The nextRun time (Fri Dec 30 23:45:02 2022) was exceeded by: {96} seconds, (1 m
12/30/2022 23:45:23 INFO  WebReports Agent: ##@=== rbmasagent3_ (Sensitive Tasks) - Multi Agent WebReport Schedule script has finish
12/30/2022 23:45:23 INFO  [3362952926660]   Done with Script
```

Figure 48

7.5 Upgrade Process

This section describes the process that occurs when the **Upgrade Versions** option is selected from the Configuration and Maintenance screen. The term “upgrade” can be confusing, but in this case, it refers to the situation that occurs when a new RBMAS CSApp has been installed on a system, with changes to the software that manages scheduling (Job Script), and the associated database entry (Registry). Not all new RBMAS versions will have changes of this nature, but when an upgrade to the Job Script and/or the Registry are required, this is identified, along with the **Upgrade Versions** option as shown in Section 1 – Miscellaneous Settings: Upgrade required after install

When the Upgrade Versions option is selected, the following message is displayed:

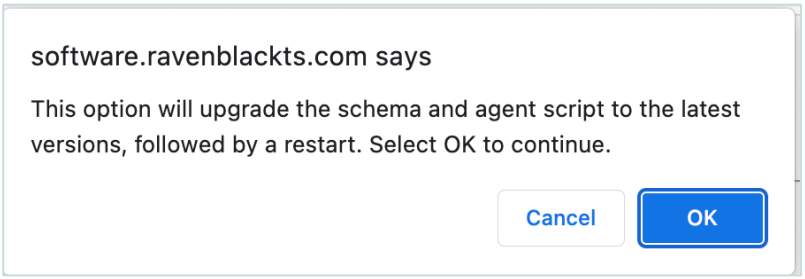


Figure 49

After selecting OK, and assuming no errors occur, this success message is presented.

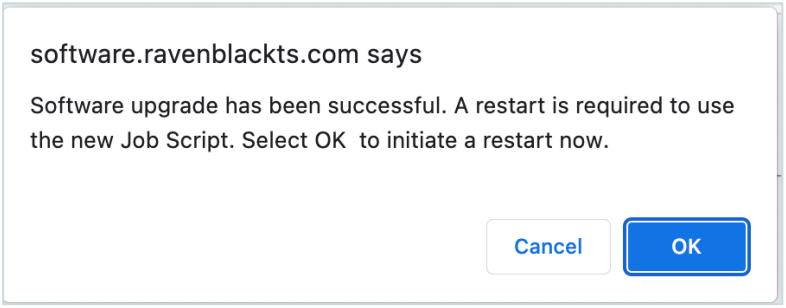


Figure 50

Selecting OK results in a restart initiation screen as shown in 4.3 Restarting the System. Once all these steps have completed, the system is now running on the latest scheduler script (Job Script) and the latest version of the Registry.

7.6 Enhanced Logging and Analysis

Once the RBMAS software is active (whether in a passive or active mode) it is possible to use two maintenance settings to control the information that is collected for WebReports being run by each agent.

Logged information is critical for scheduled WebReports as they are not interactive with a real user.

Both settings are available under the Miscellaneous settings table in *Figure 8*.

Setting Name	Value	Actions
System Status	online	Disable System Execute Restart
Performance Analysis	Inactive	Enable Analysis
Debug Logs Enabled	False	Enable Logs
Last Editor	Admin	
Last Editor Session	2022/12/02 01:49:56 Ticks:()	
Application Versions	Registry : 1.2 Job Script: 1.1.3	

Figure 51

These settings are:

- **Performance Analysis**: This setting causes the scheduler to generate detailed performance logs for every WebReport that is executed by each agent. These analysis logs are stored under **.\logs\RBAgentAnalysis**. Ravenblack support or professional services may request a zip file of this folder for analysis and reporting purposes. In a future release, an analysis tool will be deployed for customers to review this data. This feature is enabled using the **Enable Analysis** option for this setting on the **Configuration and Maintenance** page (*Figure 51*).
- **Debug Logs Enabled**: This setting causes the scheduler to generate significantly more detailed information in the agent logs to aid in debugging. More information about finding and reading log files can be found in 7.4 Interpreting Logfiles. This feature is enabled using the Enable Logs option for this setting on the **Configuration and Maintenance** (see above).

After enabling each of these settings, the settings table shows the new state, and options to disable the features:

Setting Name	Value	Actions
System Status	online	Disable System Execute Restart
Performance Analysis	Active	Disable Analysis
Debug Logs Enabled	True	Disable Logs
Last Editor	Admin	
Last Editor Session	2022/12/02 01:49:56 Ticks:()	
Application Versions	Registry : 1.2 Job Script: 1.1.3	

Figure 52

7.7 System Impacts

For the purposes of aiding support personnel, this section describes the footprint of the RBMAS software in terms of what parts of Content Server are changed or modified. Note these changes are related to a dedicated scheduling server. Future versions will allow other servers to be used.

System Component	Impact
Applications Management Volume	A Content Server Application is installed as RBMAScheduler. This is added to the csapplications folder as well as ./support/csapplications/RBMAScheduler.
.\appdata\webreports\subtags Folder	Several sub-tags are added to this folder. All sub-tags can be identified with an RB_ prefix (general sub-tags) and an RBMAS_ prefix (unique sub-tags for this application).
.\config\opentext.ini	▪ The [loader] section is modified to remove wrscheduleagent. For each RBMAS agent, an item is added using the naming convention of rbmasagentN_ (where N is numeric counting from 1). ▪ For each new agent, a section is added as [rbmasagentN_] where N is numeric. ▪ (the [wragentschedule] section remains)
,\scripts	▪ A new script (Job Script) is added to this folder called RBMAS_wrscheduleagent_run.e. Note that the original Job Script (wrscheduleagent_run.e) is left in this folder but is no longer being run while RBMAS is active.
Database KINI table	▪ RBMAScheduler is added as an IniSection to this table. This is used as a registry.

8.0 Problem Solving

This section is used to provide some useful examples of possible problems with suggested resolutions.

Currently this section has only one item in it, but it will grow as Ravenblack identifies any common issues that are not easily correctable through software fixes (e.g. intermittent system issues).

8.1 Edit Session Conflict

Symptoms:

The following message is received when saving agent data.

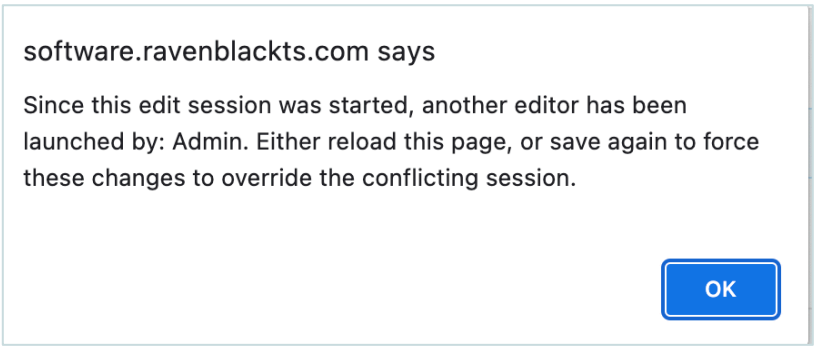


Figure 53

Cause:

The Manage WebReports Schedules interface has some logic built in to avoid the possibility of two users editing at the same time and potential overwriting each other's changes, or one user opening a second session and inadvertently losing previous changes. To manage this feature, the Manage WebReports Schedules page sets a field to show that an edit session is in progress, and who started it. This allows a message (*Figure 53*) to be generated to inform the user of the conflict with sufficient information to decide.

Resolution:

As suggested in the message. The user can choose to ignore the second session (by saving again) or to reload the page, losing their changes. If another user owns the session, this also presents an opportunity to contact them and negotiate the changes.

This will likely be a relatively rare occurrence as very few people will normally be using this page.

9.0 Terminology

Although most terms used by the RBMAS software are standard Content Server terms, some definitions are provided here to aid in following this document. (Return to Table of Contents)

Term	Description
WebReports Schedule	When a WebReport is setup, it is possible to define that the output of the WebReport is stored somewhere such as a document or sent via emails or workflows. Because these WebReports are not explicitly designed to respond to a user request, it is possible to "defer" the execution of the WebReport to a later time by creating a schedule. It is also possible to setup the WebReport to run repeatedly at pre-defined intervals. Each time the WebReport is run, it uses the identity, permissions, and privileges of the user who setup the schedule. Thus, a WebReports Schedule is made up of three parts: The ID of the WebReport to run, the ID of the user to run the WebReport, and the timing and frequency of each run.
Agent	This term refers to a single 'automated user' (single thread) running Content Server software (only handles one thing at a time) that automatically starts and runs a defined script every time the main Content Server system is restarted. The WebReports module provides its own dedicated agent to support the execution of scheduled WebReports. The default (out of the box) script used for WebReports scheduling is designed to look for WebReports that are scheduled to run, and to run them if they are due to run. With RBMAS installed, it is possible to create more than one agent dedicated to running WebReport schedules. Each agent represents a completely separate instance of the Content Server software, but with RBMAS installed, a common reference is provided so that these agents are able to coordinate the sharing of the workload, so no duplication or omission occurs. Each agent generates its own "log file" and because there is no user interaction with these WebReports, the only way to understand any outcomes or errors from the automatic execution, is by reading these log files. The naming of each file that corresponds with an agent is identified in *Figure 42*.

Thread	In Content Server the terms “thread” and “threads” are used as part of a system that allows multiple tasks to run concurrently on a given installation of Content Server. This number can be configured by an administrator. Normally each one of the threads is running a particular user request in parallel with any other user requests that are running on different threads. For agents, each Agent is an individual thread running on Content Server so if RBMAS has added 3 agents, it has also created 3 separate threads to run WebReports.
Logs/ Log files	This is a standard computer term used to specify files that contain the output messages of any executing computer process. In Content Server terms, each Thread has its own dedicated log file. Reviewing the log file provide progress messages, warning messages, error messages - any information a developer designed to be output during execution. The term “Log” by itself usually refers to each line of text in the log file
Default (agent)	The term Default refers to a specific agent that is designated with this setting. Specifically, it refers to the agent that is used automatically (by default) for any WebReports that are assigned a schedule from the WebReports destination page. It is also possible to assign a WebReport to the Default agent using the RBMAS **Manage WebReports Schedules** page. It is not possible for any user to assign to a non-default agent via the WebReports destination page as this can only be done via the RBMAS **Manage WebReports Schedules** page. Thus, any new, unique agents created using RBMAS can only be assigned by personnel who have visibility of the whole workload and the assignment strategy. Once multiple agents have been setup, any agent can be assigned as the “Default” using a control on the RBMAS **Manage WebReports Schedules** page. When RBMAS software is first enabled, there is only one agent available (using the original name: wrscheduleagent). This is automatically the “Default”. Note: this term replaces “Primary” as of RBMAS version 1.3.
Primary (agent)	This term is now obsolete as of RBMAS version 1.3 as it has been replaced by the term “Default”. It is included here in case any references to this term remain.

10.0 Beta and Future Features

This section provides information on some beta features and additional functionality that can be made available to customers.

10.1 Agent Data Analysis Tool

The Performance Analysis tool mentioned in 7.6 Enhanced Logging and Analysis is used to generate useful performance information for the active agents and the WebReports that are being run. This information is stored in files within the logs folder of Content Server. Currently the RBMAS product does not include any tools to read these files, and usually this information is provided to Ravenblack to interpret and to provide reporting. These files are readable but typically require some parsing to be useful. Ravenblack has developed a tool that can be used to interpret this information, that is currently in BETA (not currently released to customers). A version of this tool can be provided by request as a “drop-in” to customers who also have the Ravenblack Application Analyzer product. This tool will be included in future versions of RBMAS.

This diagram shows an example of this drop-in tool being used by Ravenblack to support our analysis of typical scheduling issues for future improvements.

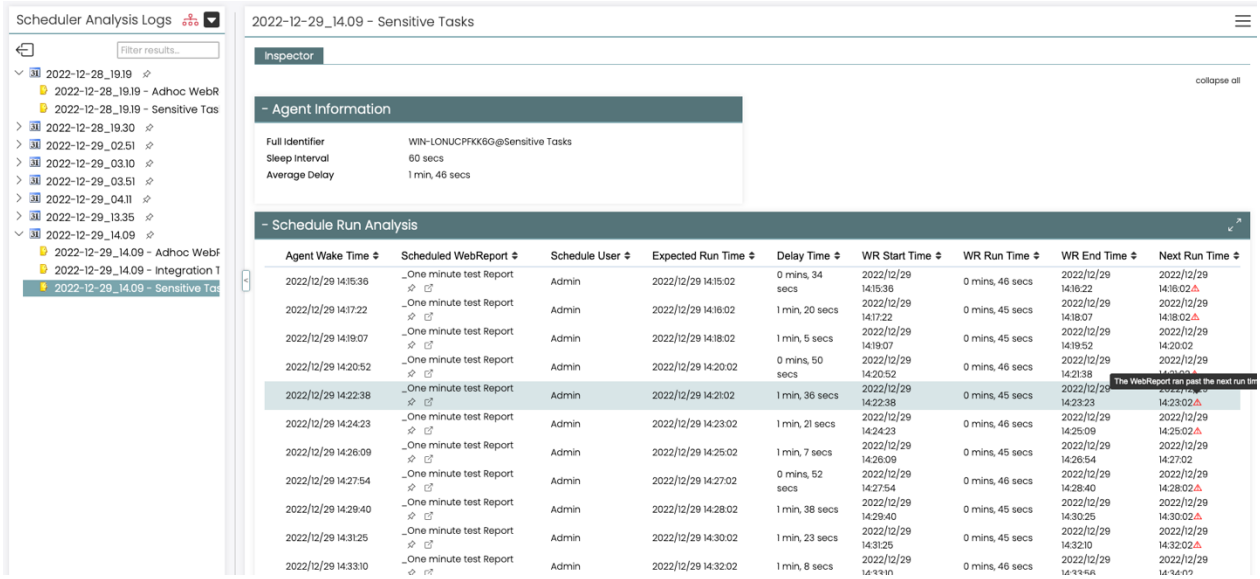


Figure 54

10.2 New Scheduling Algorithm

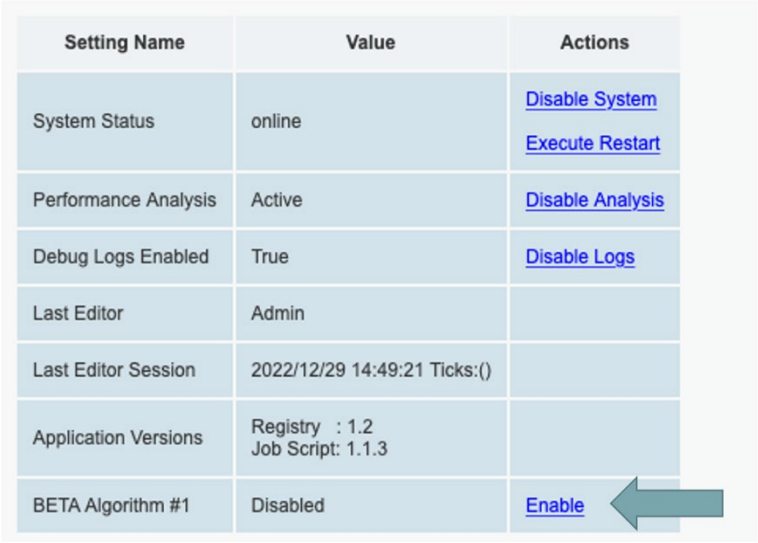
Using the performance analysis feature (along with the Agent Data Analysis Tools as described in the previous section) to analyze different scheduling cases, we have started to evolve the traditional scheduling algorithm. As of version 1.3 we have produced the first evolution of this algorithm but have not officially implemented this algorithm pending additional testing with a variety of different use cases. This new algorithm may become the default, but more likely it will be implemented as a selectable feature, allowing different algorithms for different use cases.

If a customer wants to try the new algorithm on a test system, we have made this possible through the following mechanism:

- Access the **Configuration and Maintenance** page as described above.
- Add the following parameter to the URL: &accessBETA=true, or alternatively just use this URL:

`?func=csapps.launchapp&appname=RBMAScheduler&nickname=ManageSettings&accessBETA=true`

This URL shows the standard **Configuration and Maintenance** page but includes an additional row in the Miscellaneous Settings section as shown in the following screenshot.



Setting Name	Value	Actions
System Status	online	Disable System Execute Restart
Performance Analysis	Active	Disable Analysis
Debug Logs Enabled	True	Disable Logs
Last Editor	Admin	
Last Editor Session	2022/12/29 14:49:21 Ticks:()	
Application Versions	Registry : 1.2 Job Script: 1.1.3	
BETA Algorithm #1	Disabled	Enable

Figure 55

This new algorithm is specifically aimed at scheduled tasks that need to execute on a tight interval. The algorithm reduces the amount of delay due to agent sleep time. In addition to enabling the BETA feature, this algorithm will only take effect for agents that have been set to a sleep interval of 4 minutes or less.

11.0 About Ravenblack

Ravenblack Technical Services enables users of OpenText Content Intelligence, Perspectives, and Smart View to get more out of their investments in OpenText Content Suite and Extended ECM (xECM) platforms. Owned by Greg Petti, one of the original founders of Resonate Knowledge Technologies (RKT), Ravenblack provides consulting, best practice advice, training, and development services to organizations around the world.