RAVENBLACK
TECHNICAL SERVICES

Ravenblack Products

Productivity Suite for Content Intelligence

Version: 1.1.1

Release Date: 2021-04-06

RAVENBLACK
TECHNICAL SERVICES

Productivity Suite for Content Intelligence Date: 2021-04-06

Table of Contents

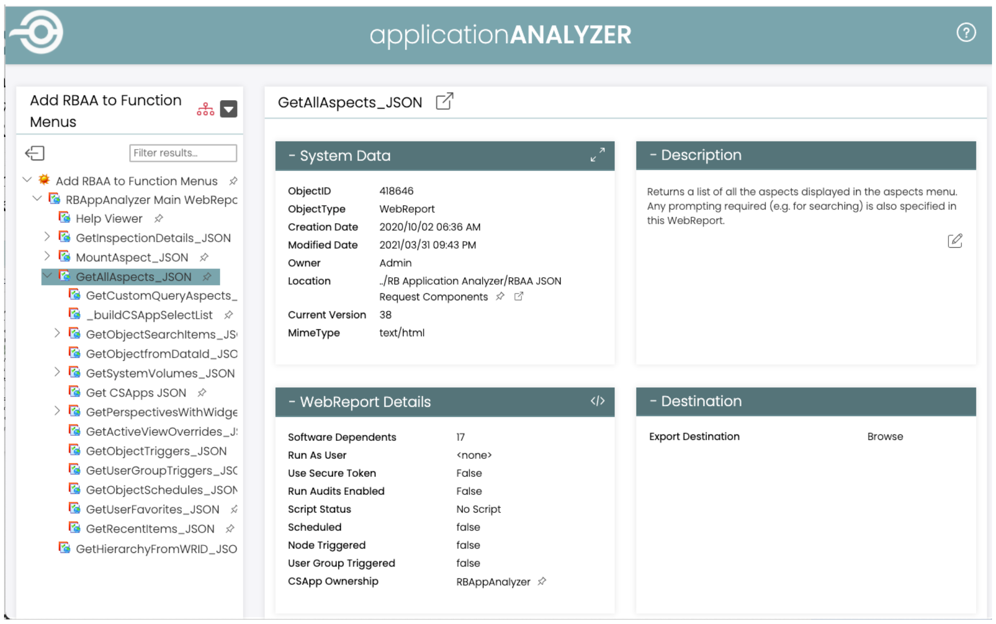
Introduction

The Ravenblack: Content Intelligence – Productivity Suite, is a set of applications and custom sub-tags that are designed to help Content Intelligence (WebReports) developers and customers with development, maintenance and support. Besides the various products bundled with this offering, this suite also includes support services and some documentation to advise on various useful methods and techniques.

Component Name	Description
Application Analyzer (RBAppAnalyzer)	This application provides (among several other features) a graphic, hierarchical view of the relationship between multiple WebReports and ActiveViews involved in a Content Server application. A fuller description can be found in the Application Analyzer section of this document.
Sub-tag loader (RBSubtagsLoad)	Dynamically populates all threads on a given server with the latest sub-tags – without a restart! Additionally, this tool allows validation and test compilation for new sub-tags, and allows the online tag guide to be re-built.
Extended Sub-tag Suite	This component provides a suite of about 30 custom sub-tags to supplement and complement the functionality currently available from OpenText.

Application Analyzer

This application allows developers and support personnel to efficiently manage applications built with Content Server components such as Perspectives, WebReports, ActiveViews, LiveReports, Forms, etc. The screen shot below shows the basic layout of the tool. The left panel provides a "finder" that displays different "aspects" of Content Server in order to find and analyze applications and application objects, and the right ("inspection") panel provides detailed information for each object, including links to any linked application objects.



Add RBAA to Function Menus
- RBAppAnalyzer Main WebRepc
  - Help Viewer
  - GetInspectionDetails_JSON
  - MountAspect_JSON
  - GetAllAspects_JSON
    - GetCustomQueryAspects_
    - _buildCSAppSelectList
    - GetObjectSearchItems_JS
    - GetObjectfromDataId_JSO
    - GetSystemVolumes_JSON
    - Get CSApps JSON
    - GetPerspectivesWithWidge
    - GetActiveViewOverrides_J
    - GetObjectTriggers_JSON
    - GetUserGroupTriggers_JSC
    - GetObjectSchedules_JSON
    - GetUserFavorites_JSON
    - GetRecentItems_JSON
  - GetHierarchyFromWRID_JSO

- System Data

ObjectID	418646
ObjectType	WebReport
Creation Date	2020/10/02 06:36 AM
Modified Date	2021/03/31 09:43 PM
Owner	Admin
Location	../RB Application Analyzer/RBAA JSON Request Components
Current Version	38
MimeType	text/html

- Description

Returns a list of all the aspects displayed in the aspects menu. Any prompting required (e.g. for searching) is also specified in this WebReport.

- WebReport Details

Software Dependents	17
Run As User	<none>
Use Secure Token	False
Run Audits Enabled	False
Script Status	No Script
Scheduled	false
Node Triggered	false
User Group Triggered	false
CSApp Ownership	RBAppAnalyzer

- Destination

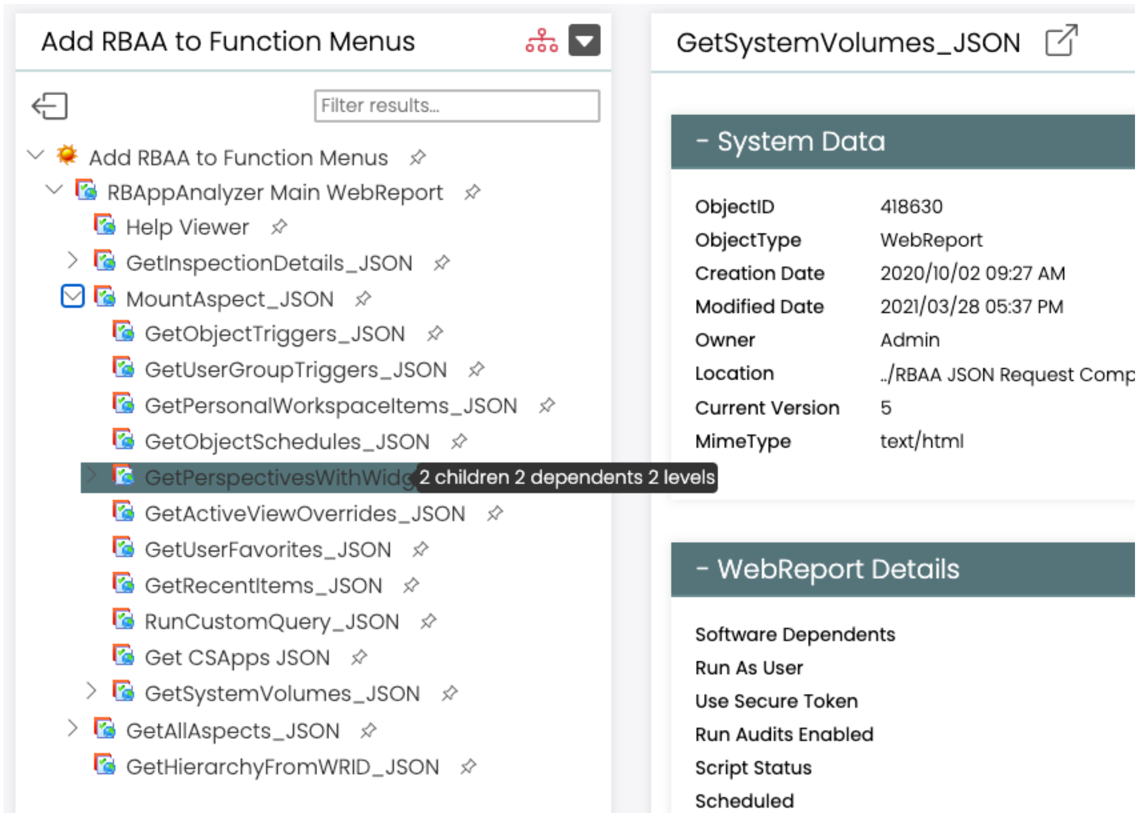
Export Destination	Browse

Software Relationship Analysis

At the core of this application is the ability to identify and display the many interactions and dependencies that exist in a typical WebReports based application. This function is implemented by "mounting" any WebReport in the finder which initiates analysis of all the source code for each WebReport, building a visual interpretation of all the software links.

There are two different visuals available to support this feature as shown in the following two screen shots.

Vertical Visualization of Software Dependencies


Add RBAA to Function Menus

Filter results...

- Add RBAA to Function Menus
  - RBAppAnalyzer Main WebReport
    - Help Viewer
    - GetInspectionDetails_JSON
    - MountAspect_JSON
      - GetObjectTriggers_JSON
      - GetUserGroupTriggers_JSON
      - GetPersonalWorkspaceItems_JSON
      - GetObjectSchedules_JSON
      - GetPerspectivesWithWidg 2 children 2 dependents 2 levels
      - GetActiveViewOverrides_JSON
      - GetUserFavorites_JSON
      - GetRecentItems_JSON
      - RunCustomQuery_JSON
      - Get CSApps JSON
      - GetSystemVolumes_JSON
    - GetAllAspects_JSON
    - GetHierarchyFromWRID_JSON

GetSystemVolumes_JSON

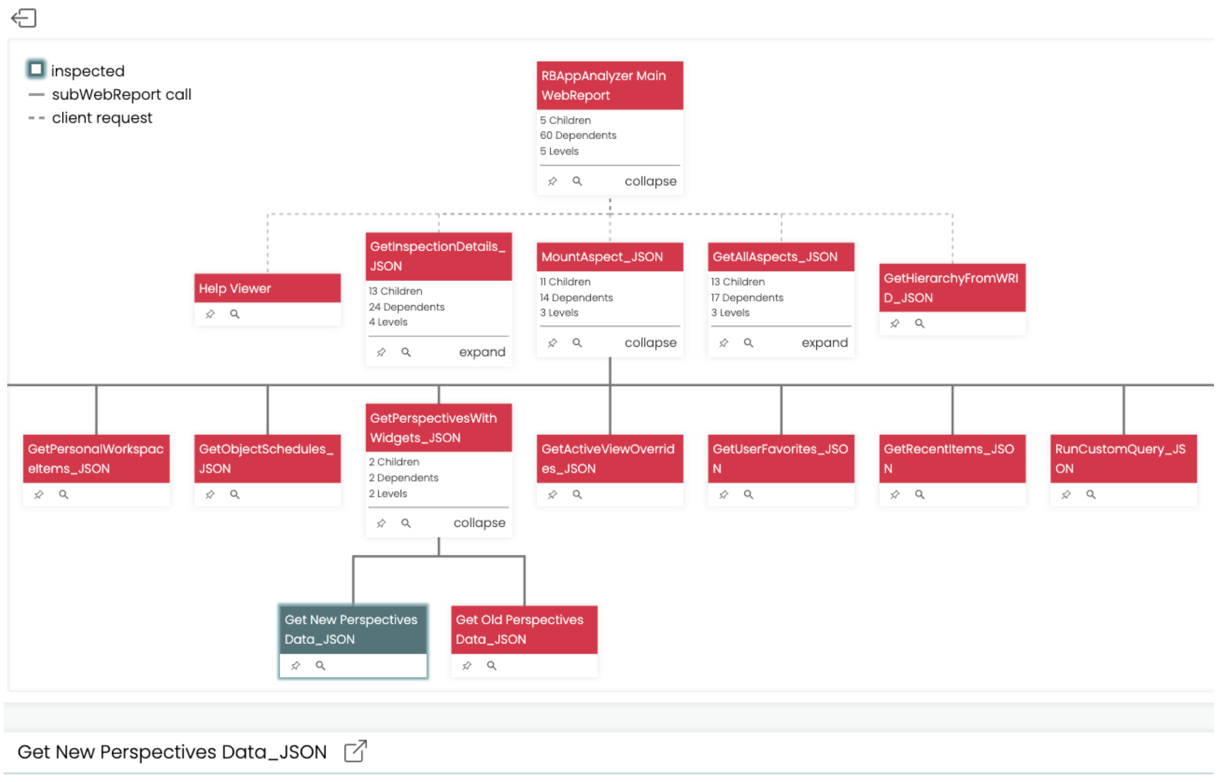
- System Data

ObjectID	418630
ObjectType	WebReport
Creation Date	2020/10/02 09:27 AM
Modified Date	2021/03/28 05:37 PM
Owner	Admin
Location	../RBAA JSON Request Comp
Current Version	5
MimeType	text/html

- WebReport Details

Software Dependents
Run As User
Use Secure Token
Run Audits Enabled
Script Status
Scheduled

Horizontal Visualization of Software Dependencies (Graph)

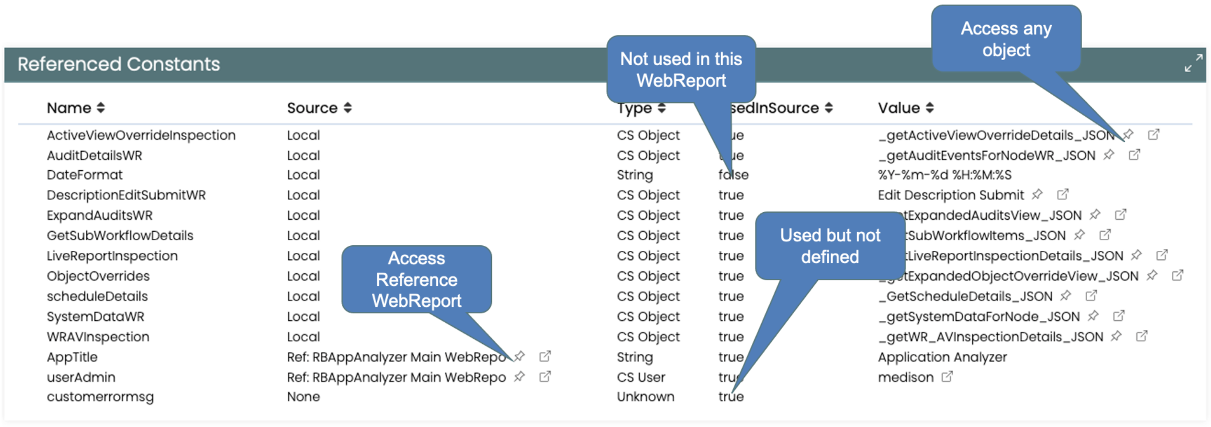


Inspection Panel

From the finder views, any node can be selected in order to reveal all of its relevant details and features in the inspection panel". The inspection panel is divided into "sections" and shows at a glance, all of the major attributes and features of the selected object in one place. For WebReports, this includes features such as whether it is used in a Perspective, whether it is scheduled or triggered and how many software dependents it has. Additionally, for a WebReport or an ActiveView, specific sections like Data Source, Destination, Constants and Parameters are presented with unique, useful information that includes hyper-links for each dependent object. You can choose to either mount these objects (open them within the finder as a root object) or to execute classic Content Server actions for each object.

Constants

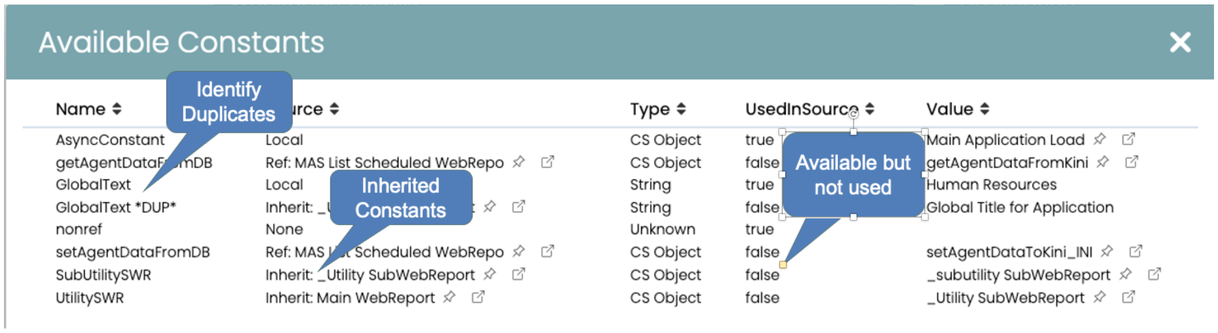
This inspection panel is particularly useful for WebReports Constants where the display not only shows locally defined constants, but identifies which ones are actually used in the source code. This removes the need to search through source code, open tabs and select function menus to trace referenced objects. The clear, visibility of values even when they are not defined locally, along with hyperlinks to the inherited and referenced WebReports makes tracking and managing constant values significantly easier. Additionally, we can see when we are using constants that have no definitions.



Referenced Constants

Name	Source	Type	sedInSource	Value
ActiveViewOverrideInspection	Local	CS Object	ue	_getActiveViewOverrideDetails_JSON
AuditDetailsWR	Local	CS Object	ue	_getAuditEventsForNodeWR_JSON
DateFormat	Local	String	false	%Y-%m-%d %H:%M:%S
DescriptionEditSubmitWR	Local	CS Object	true	Edit Description Submit
ExpandAuditsWR	Local	CS Object	true	tExpandedAuditsView_JSON
GetSubWorkflowDetails	Local	CS Object	true	SubWorkflowItems_JSON
LiveReportInspection	Local	CS Object	true	LiveReportInspectionDetails_JSON
ObjectOverrides	Local	CS Object	true	tExpandedObjectOverrideView_JSON
scheduleDetails	Local	CS Object	true	_GetScheduleDetails_JSON
SystemDataWR	Local	CS Object	true	_getSystemDataForNode_JSON
WRAVInspection	Local	CS Object	true	_getWR_AVInspectionDetails_JSON
AppTitle	Ref: RBAppAnalyzer Main WebRepo	String	true	Application Analyzer
userAdmin	Ref: RBAppAnalyzer Main WebRepo	CS User	tru	medison
customerrormsg	None	Unknown	true	

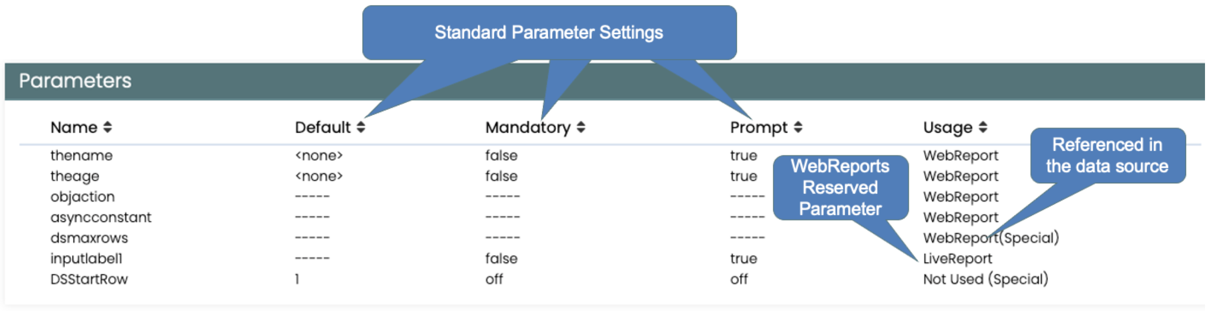
The constants section can also be expanded to show ALL available constants whether they are used or not, making it easy to see duplication, and architecture inefficiencies. The example below illustrates some of the observations that this screen can provide.



Name	rce	Type	UsedInSource	Value
AsyncConstant	Local	CS Object	true	Main Application Load
getAgentDataF[illegible]mDB	Ref: MAS List Scheduled WebRepo	CS Object	false	getAgentDataFromKini
GlobalText	Local	String	true	Human Resources
GlobalText *DUP*	Inherit: _[illegible]	String	false	Global Title for Application
nonref	None	Unknown	true	
setAgentDataFromDB	Ref: MAS [illegible] Scheduled WebRepo	CS Object	false	setAgentDataToKini_INI
SubUtilitySWR	Inherit: _Utility SubWebReport	CS Object	false	_subutility SubWebReport
UtilitySWR	Inherit: Main WebReport	CS Object	false	_Utility SubWebReport

Parameters

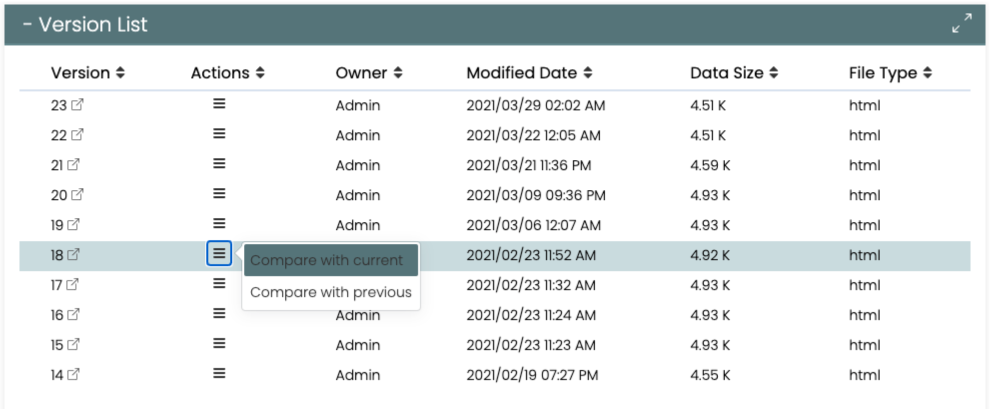
This section of the inspection panel provides some very useful extensions to the information that is available through the conventional WebReports feature set. Besides showing some of the normal settings from the parameters tab, this section shows whether parameters are being referenced in the WebReports code, as parameters for the data source (a LiveReport in this example) or whether they have been defined by a Perspective (if the WebReport is being used as a widget). Additionally, if the source code of the WebReport, or its parameters tab, references any of the special, reserved WebReports parameters, this is also identified.


Standard Parameter Settings
Parameters
WebReports Reserved Parameter
Referenced in the data source

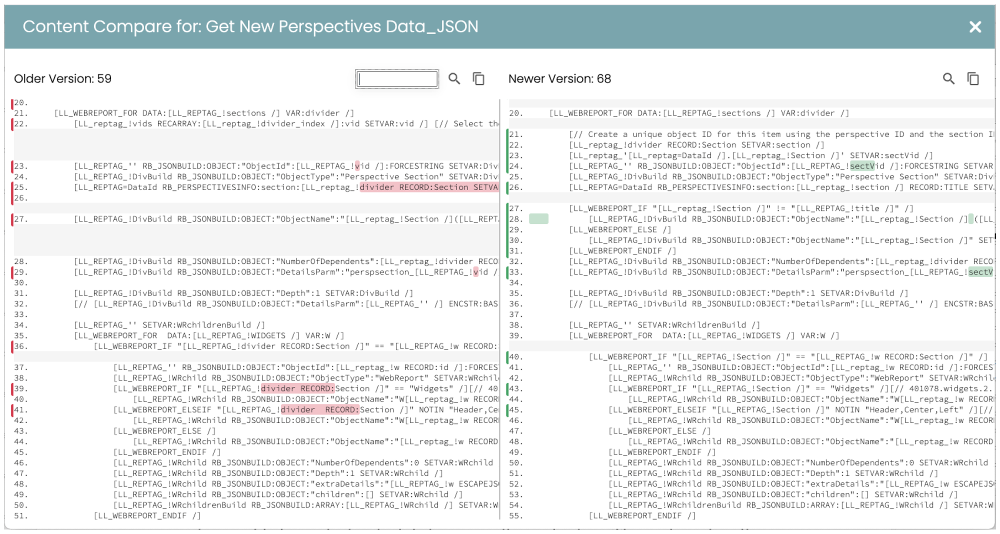
Name	Default	Mandatory	Prompt	Usage
thename	<none>	false	true	WebReport
theage	<none>	false	true	WebReport
objaction	-----	-----	-----	WebReport
asyncconstant	-----	-----	-----	WebReport
dsmaxrows	-----	-----	-----	WebReport(Special)
inputlabel1	-----	false	true	LiveReport
DSStartRow	1	off	off	Not Used (Special)

Version List

This section lists the last 10 versions (100 in the expanded view), allowing any version to be compared with the current version or the previous version using a “diff” tool as shown in the second screen shot.



Version	Actions	Owner	Modified Date	Data Size	File Type
23	≡	Admin	2021/03/29 02:02 AM	4.51 K	html
22	≡	Admin	2021/03/22 12:05 AM	4.51 K	html
21	≡	Admin	2021/03/21 11:36 PM	4.59 K	html
20	≡	Admin	2021/03/09 09:36 PM	4.93 K	html
19	≡	Admin	2021/03/06 12:07 AM	4.93 K	html
18	≡		2021/02/23 11:52 AM	4.92 K	html
17	≡		2021/02/23 11:32 AM	4.93 K	html
16	≡	Admin	2021/02/23 11:24 AM	4.93 K	html
15	≡	Admin	2021/02/23 11:23 AM	4.93 K	html
14	≡	Admin	2021/02/19 07:27 PM	4.55 K	html



Older Version: 59

```

    [LL_WEBREPORT_FOR DATA:[LL_REPTAG_!sections /] VAR:divider /]
        [LL_reptag_!vids RECARRAY:[LL_reptag_!divider_index /]:vid SETVAR:vid /] [// Select th
        [LL_REPTAG_'' RB_JSONBUILD:OBJECT:"ObjectId":[LL_REPTAG_!vid /]:FORCESTRING SETVAR:Div
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"ObjectType":"Perspective Section" SETVAR:Div
        [LL_REPTAG=DataId RB_PERSPECTIVESINFO:section:[LL_reptag_!divider RECORD:Section SETVA

        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"ObjectName":"[LL_reptag_!Section /]([LL_REPT
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"NumberOfDependents":[LL_reptag_!divider RECO
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"DetailsParm":"perspsection_[LL_REPTAG_!vid /

        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"Depth":1 SETVAR:DivBuild /]
        [// [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"DetailsParm":[LL_REPTAG_'' /] ENCSTR:BAS

        [LL_REPTAG_'' SETVAR:WRchildrenBuild /]
        [LL_WEBREPORT_FOR  DATA:[LL_REPTAG_!WIDGETS /] VAR:W /]
            [LL_WEBREPORT_IF "[LL_REPTAG_!divider RECORD:Section /]" == "[LL_REPTAG_!w RECORD:
                [LL_REPTAG_'' RB_JSONBUILD:OBJECT:"ObjectId":[LL_reptag_!w RECORD:id /]:FORCES
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectType":"WebReport" SETVAR:WRchil
                [LL_WEBREPORT_IF "[LL_REPTAG_!divider RECORD:Section /]" == "Widgets" /][// 40
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"W[LL_reptag_!w RECOR
                [LL_WEBREPORT_ELSEIF "[LL_REPTAG_!divider  RECORD:Section /]" NOTIN "Header,Ce
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"W[LL_reptag_!w RECOR
                [LL_WEBREPORT_ELSE /]
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"[LL_reptag_!w RECORD
                [LL_WEBREPORT_ENDIF /]
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"NumberOfDependents":0 SETVAR:WRchild
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"Depth":1 SETVAR:WRchild /]
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"extraDetails":"[LL_REPTAG_!w ESCAPEJS
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"children":[] SETVAR:WRchild /]
                [LL_REPTAG_!WRchildrenBuild RB_JSONBUILD:ARRAY:[LL_REPTAG_!WRchild /] SETVAR:W
            [LL_WEBREPORT_ENDIF /]
```

Newer Version: 68

```
    [LL_WEBREPORT_FOR DATA:[LL_REPTAG_!sections /] VAR:divider /]
        [// Create a unique object ID for this item using the perspective ID and the section I
        [LL_reptag_!divider RECORD:Section SETVAR:section /]
        [LL_reptag_'[LL_reptag=DataId /].[LL_reptag_!Section /]' SETVAR:sectVid /]
        [LL_REPTAG_'' RB_JSONBUILD:OBJECT:"ObjectId":[LL_REPTAG_!sectVid /]:FORCESTRING SETVAR
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"ObjectType":"Perspective Section" SETVAR:Div
        [LL_REPTAG=DataId RB_PERSPECTIVESINFO:section:[LL_reptag_!section /] RECORD:TITLE SETV

        [LL_WEBREPORT_IF "[LL_reptag_!Section /]" != "[LL_REPTAG_!title /]" /]
            [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"ObjectName":"[LL_reptag_!Section /] ([LL
        [LL_WEBREPORT_ELSE /]
            [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"ObjectName":"[LL_reptag_!Section /]" SET
        [LL_WEBREPORT_ENDIF /]
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"NumberOfDependents":[LL_reptag_!divider RECO
        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"DetailsParm":"perspsection_[LL_REPTAG_!sectV

        [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"Depth":1 SETVAR:DivBuild /]
        [// [LL_REPTAG_!DivBuild RB_JSONBUILD:OBJECT:"DetailsParm":[LL_REPTAG_'' /] ENCSTR:BAS

        [LL_REPTAG_'' SETVAR:WRchildrenBuild /]
        [LL_WEBREPORT_FOR  DATA:[LL_REPTAG_!WIDGETS /] VAR:W /]

            [LL_WEBREPORT_IF "[LL_REPTAG_!Section /]" == "[LL_REPTAG_!w RECORD:Section /]" /]
                [LL_REPTAG_'' RB_JSONBUILD:OBJECT:"ObjectId":[LL_reptag_!w RECORD:id /]:FORCES
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectType":"WebReport" SETVAR:WRchil
                [LL_WEBREPORT_IF "[LL_REPTAG_!Section /]" == "Widgets" /][// 401078.widgets.2.
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"W[LL_reptag_!w RECOR
                [LL_WEBREPORT_ELSEIF "[LL_REPTAG_!Section /]" NOTIN "Header,Center,Left" /][//
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"W[LL_reptag_!w RECOR
                [LL_WEBREPORT_ELSE /]
                    [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"ObjectName":"[LL_reptag_!w RECORD
                [LL_WEBREPORT_ENDIF /]
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"NumberOfDependents":0 SETVAR:WRchild
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"Depth":1 SETVAR:WRchild /]
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"extraDetails":"[LL_REPTAG_!w ESCAPEJS
                [LL_REPTAG_!WRchild RB_JSONBUILD:OBJECT:"children":[] SETVAR:WRchild /]
                [LL_REPTAG_!WRchildrenBuild RB_JSONBUILD:ARRAY:[LL_REPTAG_!WRchild /] SETVAR:W
            [LL_WEBREPORT_ENDIF /]
```

Description/Documentation

The inspection panel includes a description section for each object. This makes any commenting visible for each object while traversing the application. This section also allows easy editing of this description to allow incremental documentation (in a future version it will be possible to generate documentation based on these fields).

Finding Applications

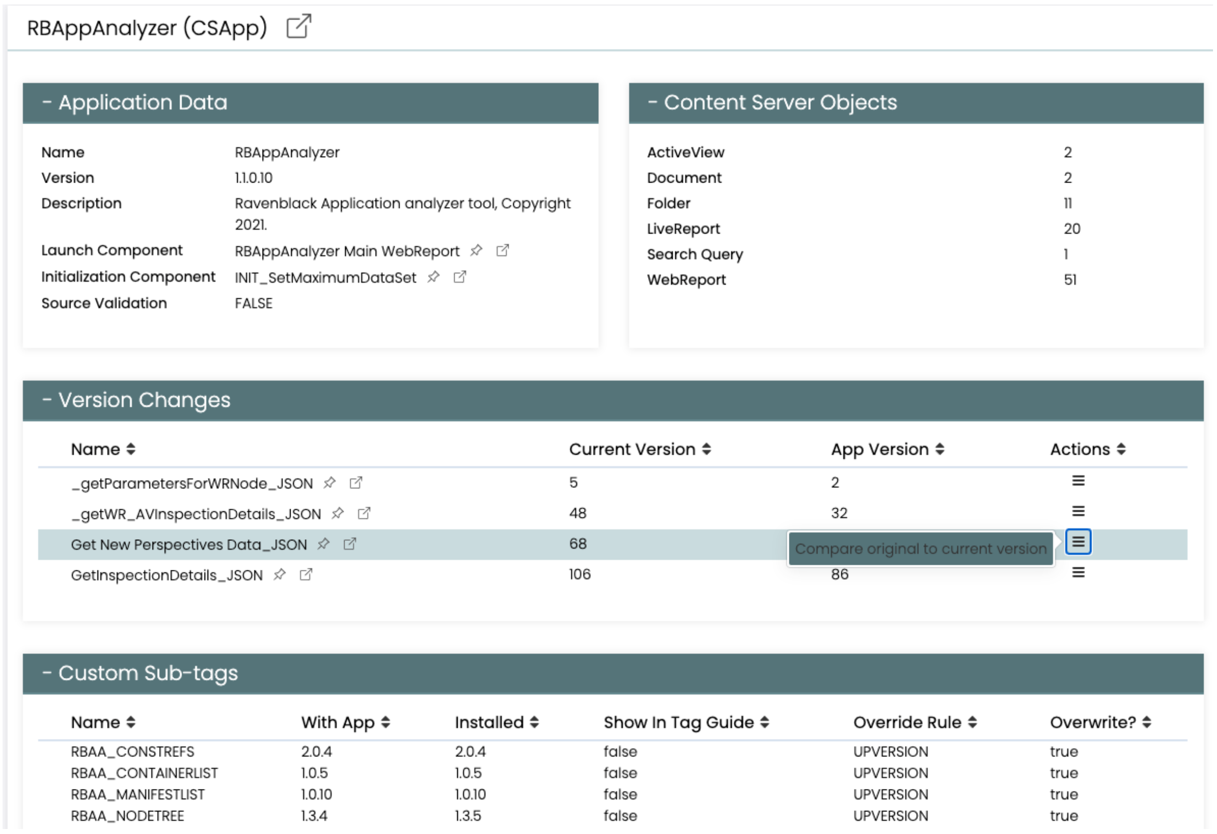
In order to take advantage of the various features that aid in analyzing WebReports, it is important to be able to find applications and their associated objects as easily as possible. To facilitate this, we have included a feature that allows the system to be browsed using various useful paradigms (aspects). These can be considered as different lists, reports, searches or browsing tools. The current aspects supported are:

- Searching – currently two pre-defined searches are supported. The main search allows full text searching by key words as well as document type and application ownership.
- System browsing – traditional container-based browsing through a tree or hierarchy to find applications or application objects that are stored in Content Server containers.
- Content Server Applications – provides a list of all the logical CSApps, regardless of whether the components are stored in the same container or not. This feature also provides a useful inspection panel for CSApps that displays the various parts of these applications.
- Triggered WebReports – shows all WebReports on a system that are being run from an object or user group trigger (with the triggering objects). This particular view is a very unique feature as there is no other simple way to see these WebReports.
- Scheduled WebReports – this view provides information similar to the admin page for scheduled WebReports but provide a means to Inspect and analyze these WebReports. .
- Perspectives with WebReport Widgets – returns all perspectives that use WebReports widgets along with all of these (WebReport) widgets for selection and inspection.
- ActiveView Overrides – returns all ActiveViews that are actively used as overrides in classic Content Server, along with all corresponding overrides and related override objects.
- Favourites & Recent Items – two different browse types that allow the user to create access to their most commonly accessed items.

This list of browse types will continue to expand, and it is possible to add on custom aspects simply based on Search Queries or LiveReports. Additionally, custom WebReports components can be created (with some services from Ravenblack).

Content Server Application Details

From the CS Applications view, you will see a list of applications on the system. If you select one of these applications, you can see at a glance some of the key features of that application. You can "mount" the application as the root object for analysis by double clicking on the application, and from this view you could also select and mount the "launch component" for the application providing the software architecture at a glance using the vertical or hierarchical view.
Note the "Version Changes" section allows "diff" comparisons for any changed objects.



RBAppAnalyzer (CSApp)

- Application Data

Name	RBAppAnalyzer
Version	1.1.0.10
Description	Ravenblack Application analyzer tool, Copyright 2021.
Launch Component	RBAppAnalyzer Main WebReport
Initialization Component	INIT_SetMaximumDataSet
Source Validation	FALSE

- Content Server Objects

ActiveView	2
Document	2
Folder	11
LiveReport	20
Search Query	1
WebReport	51

- Version Changes

Name	Current Version	App Version	Actions
_getParametersForWRNode_JSON	5	2	≡
_getWR_AVInspectionDetails_JSON	48	32	≡
Get New Perspectives Data_JSON	68		≡
GetInspectionDetails_JSON	106	86	≡

- Custom Sub-tags

Name	With App	Installed	Show In Tag Guide	Override Rule	Overwrite?
RBAA_CONSTREFS	2.0.4	2.0.4	false	UPVERSION	true
RBAA_CONTAINERLIST	1.0.5	1.0.5	false	UPVERSION	true
RBAA_MANIFESTLIST	1.0.10	1.0.10	false	UPVERSION	true
RBAA_NODETREE	1.3.4	1.3.5	false	UPVERSION	true

Application Launch & Integration

This application comes with a selection of ActiveView overrides to allow easy (optional) integration of the tool into Content Server. This integration is normally conditionally controlled so that only sysadmin users will ever see the tool. With this integration enabled, the application can be initiated in a few different ways:

- From the Admin menu, selecting **Open Application Analyzer** opens the application with no objects in focus. The application/object finder could then be used to find an application or object to focus on.
- From the function menu of various objects in Content Server, you can select: **Open Application Analyzer** and the application opens with that object already mounted as a root object in the finder panel (in focus).
- From the Application container for **RB Application Analyzer**, you can select the main WebReport: **RBAppAnalyzer Main WR** and the application loads with no objects in focus.
- From anywhere in the system, the URL: **?func=csapps.launchapp&appname=RBAppAnalyzer** can be used to launch the application.

Miscellaneous

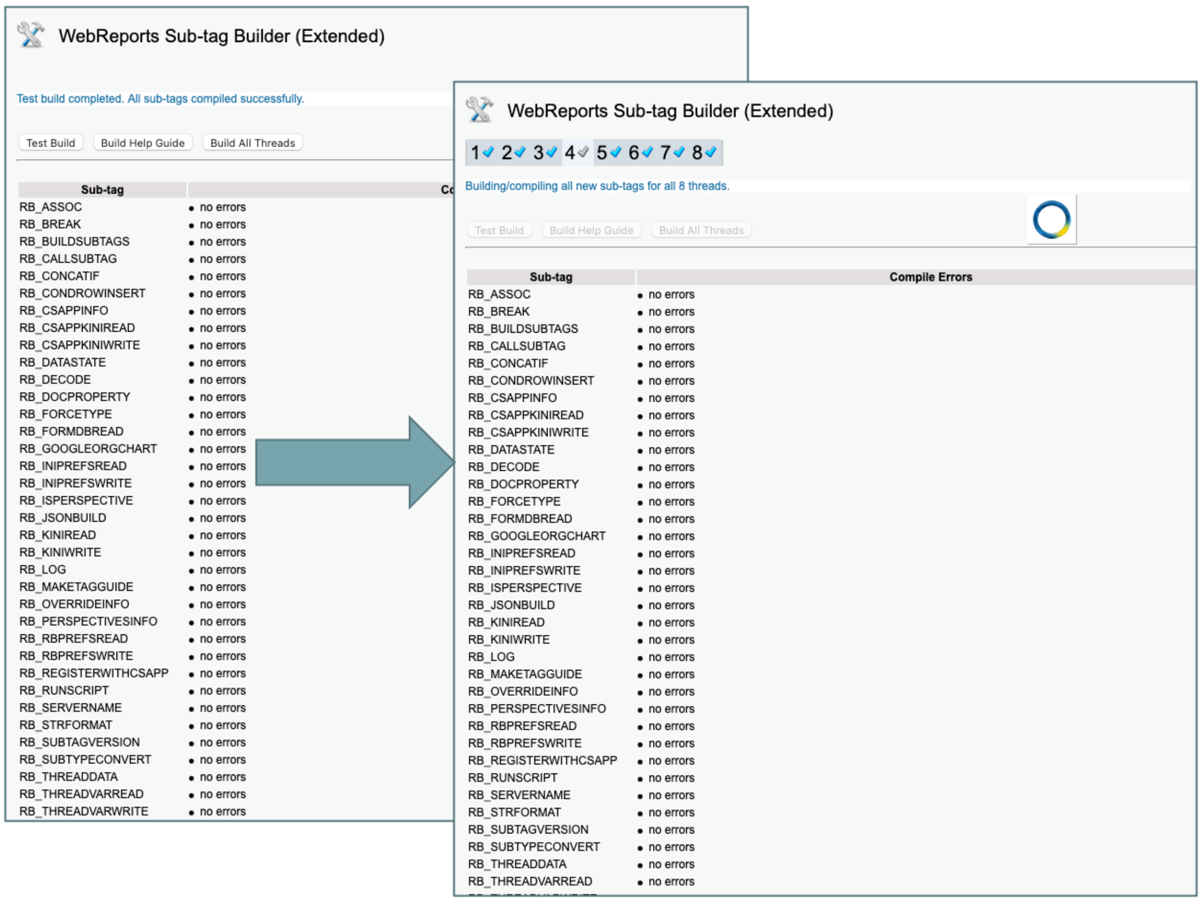
This application is run using a JavaScript framework loaded via a main WebReport, and multiple supporting data WebReports that return data to the application. It is installed as a Content Server Application and it can be run within the classic architecture with the top menu optionally visible. This application has a set of application specific sub-tags included but also uses several of the sub-tags that are included in the Ravenblack extended sub-tag suite (mentioned later in this document).

Sub-tag Loader Tool

(No restart required)

This tool provides an enhanced version of the standard sub-tag builder screen.

With this screen you can initiate a build process that causes the tool to build any new (drop-in) sub-tags for each thread on the server without performing any restarts.
This tool can also be useful to test thread availability. If any threads are blocked or tied up, this situation would be visible via the thread population graphic (shown in this screen shot).



Test build completed. All sub-tags compiled successfully.

Sub-tag	Co
RB_ASSOC	● no errors
RB_BREAK	● no errors
RB_BUILDSUBTAGS	● no errors
RB_CALLSUBTAG	● no errors
RB_CONCATIF	● no errors
RB_CONDROWINSERT	● no errors
RB_CSAPPINFO	● no errors
RB_CSAPPKINIREAD	● no errors
RB_CSAPPKINIWRITE	● no errors
RB_DATASTATE	● no errors
RB_DECODE	● no errors
RB_DOCPROPERTY	● no errors
RB_FORCETYPE	● no errors
RB_FORMDBREAD	● no errors
RB_GOOGLEORGCHART	● no errors
RB_INIPREFSREAD	● no errors
RB_INIPREFSWRITE	● no errors
RB_ISPERSPECTIVE	● no errors
RB_JSONBUILD	● no errors
RB_KINIREAD	● no errors
RB_KINIWRITE	● no errors
RB_LOG	● no errors
RB_MAKETAGGUIDE	● no errors
RB_OVERRIDEINFO	● no errors
RB_PERSPECTIVESINFO	● no errors
RB_RBPREFSREAD	● no errors
RB_RBPREFSWRITE	● no errors
RB_REGISTERWITHCSAPP	● no errors
RB_RUNSCRIPT	● no errors
RB_SERVERNAME	● no errors
RB_STRFORMAT	● no errors
RB_SUBTAGVERSION	● no errors
RB_SUBTYPECONVERT	● no errors
RB_THREADDATA	● no errors
RB_THREADVARREAD	● no errors
RB_THREADVARWRITE	● no errors

Building/compiling all new sub-tags for all 8 threads.

Sub-tag	Compile Errors
RB_ASSOC	● no errors
RB_BREAK	● no errors
RB_BUILDSUBTAGS	● no errors
RB_CALLSUBTAG	● no errors
RB_CONCATIF	● no errors
RB_CONDROWINSERT	● no errors
RB_CSAPPINFO	● no errors
RB_CSAPPKINIREAD	● no errors
RB_CSAPPKINIWRITE	● no errors
RB_DATASTATE	● no errors
RB_DECODE	● no errors
RB_DOCPROPERTY	● no errors
RB_FORCETYPE	● no errors
RB_FORMDBREAD	● no errors
RB_GOOGLEORGCHART	● no errors
RB_INIPREFSREAD	● no errors
RB_INIPREFSWRITE	● no errors
RB_ISPERSPECTIVE	● no errors
RB_JSONBUILD	● no errors
RB_KINIREAD	● no errors
RB_KINIWRITE	● no errors
RB_LOG	● no errors
RB_MAKETAGGUIDE	● no errors
RB_OVERRIDEINFO	● no errors
RB_PERSPECTIVESINFO	● no errors
RB_RBPREFSREAD	● no errors
RB_RBPREFSWRITE	● no errors
RB_REGISTERWITHCSAPP	● no errors
RB_RUNSCRIPT	● no errors
RB_SERVERNAME	● no errors
RB_STRFORMAT	● no errors
RB_SUBTAGVERSION	● no errors
RB_SUBTYPECONVERT	● no errors
RB_THREADDATA	● no errors
RB_THREADVARREAD	● no errors

Extended Sub-Tag Suite

The following table provides a list of all the sub-tags that are currently in the Ravenblack library. These sub-tags have been divided into 4 different categories as per the tables below.

Development, Support and Debugging Sub-tags

These sub-tags are predominantly useful to aid in debugging or maintenance activities.

Sub-tag Name	Description
RB_Break	Creates a breakpoint in the script that is running the current WebReport. This sub-tag is only useful when Eclipse or builder are available and connected. When the break occurs, this sub-tag has pre-setup two variables to make the current sub-tag data and variables easier to see.
RB_BuildSubtags	Builds the current set of sub-tag drop-ins. Allows the build to be conditional on having not been run since the last restart. Also allows all of the output from the build to be returned.
RB_RegisterWithCSApp	Used to mark any given WebReport as being owned by a particular Content Server Application (CSApp). Ownership is normally established during install but this sub-tag allows developers to create this ownership relationship during development.
RB_MakeTagGuide	Forces re-building of the tag guide. This is useful when a new sub-tag has been dropped in and the developer wants new help to be bound into the tag guide.
RB_ThreadData	Returns thread information such as the number of threads on a server and the current thread number.
RB_Timer	Creates a comprehensive timing framework by capturing the timestamps between multiple points of execution and output in the WebReport, thread log, or via the RB_logs mechanism.
RB_Trace	Generates a stack trace (trace log) from any point in a WebReport.

General Application Development Sub-tags

These sub-tags are designed to provide useful functionality for developers.

Sub-tag Name	Description
RB_ConcatIf	Provides a variation on the DECODE or RB_Decode sub-tags. Rather than replacing the original value with a new value when a match is found, this sub-tag pre-pends or appends a new value to the original value. Supports all of the RB_Decode functions.
RB_CondRowInsert	Outputs data strings based on a few defined row conditions such as first row, not first row, last row, not last row. Useful for building lists in the row section where the syntax is sensitive to opening, closing and separating characters.
RB_ForceType	Forces Oscript types from string form, into their native types. This is useful in some scenarios where other sub-tags are not detecting types correctly. It can also be used to force a native type into a string.
RB_JSONBuild	Allows building of JSON structures with objects and arrays. Automatically adds new values to each structure, modifying the syntax as required.
RB_Log	Provides custom logging functionality. Log files can be created and written to from any point in any WebReport. Allows customized messages or status information to be written to the output, the thread log, or to a custom log file specified through the sub-tag. This can be used to aid in debugging or to enhance a developed application.
RB_ServerName	Returns the full host name of a server from the OpenText “System” package or the server name from the Opentext.ini file.
RB_StrFormat	Provides a string format function (equivalent to STR.Format in OScript or functions like printf etc. in other languages).
RB_SubtypeConvert	Allows multiple useful pieces of sub-type information to be retrieved based on DataId, SubType Number or SubType Name as inputs. Possible outputs include: subtype name, sub-type number, icon path and properties such as “isContainer”, “isVolume”, etc.
RB_Decode	Provides an enhanced version of the DECODE sub-tag. Adds the capability to use comparison operators (e.g. <=,<,>,>=) as well as type testing such as “isNumber” and “IN” testing for lists and strings.
RB_ DocPropertiesRead	This sub-tag returns document properties for appropriate Content Server documents. (For example, Microsoft Office documents, PDF, etc.)

Data Saving & Retrieval Sub-tags

This group of sub-tags provides the ability to read or write from all levels of the system: threads, servers, and the database.

Sub-tag Name	Description
RB_CSAppKiniRead	Allows KINI entries for a given Content Server Application (CSAapp) to be read.
RB_CSAppKiniWrite	Allows KINI entries for a given CSApp to be added, edited or deleted. Note, this (and the corresponding Read sub-tag) provide a relatively safe, contained way to utilize the KINI table for custom applications.
RB_FormDBRead	Allows values to be read from an OpenText Forms, SQL table. This is similar to some core sub-tags but with useful enhancements such as the ability to form more complex queries to return data.
RB_FormDBWrite	Allows values to be added to, edited, or deleted from an OpenText forms, SQL table. Similar to FormDBAction but it allows any column to be used to find a row and has the potential for multiple rows to be edited or deleted.
RB_INIPrefsRead	Allows any setting to be read from the Opentext.ini file. Note, any WebReport running this sub-tag must have a sysadmin privilege level.
RB_INIPrefsWrite	Allows any setting to be added to, edited or deleted from the Opentext.ini file. Note, any WebReport running this sub-tag must have a sysadmin privilege level.
RB_KiniRead	Allows any entry to be read from the Kini table. Note, any WebReport running this sub-tag must have a sysadmin privilege level.
RB_KiniWrite	Allows any entry to be to be added to, edited or deleted from the Kini table. Note, any WebReport running this sub-tag must have a sysadmin privilege level.
RB_RBPrefsRead	Allows entries to be read from the ravenblack.ini file.
RB_RBPrefsWrite	Allows entries to be written, modified, or deleted from the ravenblack.ini file. This sub-tag will create the Ravenblack.ini file if it doesn't already exist.

RB_ThreadVarRead	Allows data to be read from a "Thread variable" (unique to a given thread).
RB_ThreadVarWrite	Allows data to be written to a "Thread Variable" (unique to a given thread).

Extensions to Existing Sub-tags

This section represents any sub-tags that are overrides of an existing sub-tag. Generally, we provide new "RB_" versions of our sub-tags to avoid any contention with future OpenText development; however, sometimes we provide sub-tags that replace the current OpenText version.

Sub-tag Name	**Description**
LLURL_ FUNCTIONMENU	Extends standard LLURL to add the ability to create a function menu for a specific version of an object (standard LLURL only works with current version).
RB_VERSIONACTION	Enhancement to allow MIMETYPE, FILETYPE, and FILENAME to be updated for any given version (standard version only supports description).

About Ravenblack

Ravenblack Technical Services enables users of OpenText Content Intelligence (WebReports, ActiveView, etc.), 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.