



Rsam 9.2

Performance Guidance for Rsam Administrators

Version: 1.4 | May 2018

Rsam © 2018. All rights reserved

[Privacy Policy](#) | [Terms of Service](#)

Contents

- About this Document..... 4
 - Typical Customer Performance 4
 - Most Common Causes of Performance Issues 4
 - Cloud Customers vs On-Prem Customers..... 4
- System Considerations (on-prem customers) 5
 - System Requirements 5
 - Physical vs Virtual Machines 5
 - Dedicated vs Shared Servers 6
 - Importance of Fast Disks with SQL..... 6
 - Separation of SQL Server and Web Server 6
 - Splitting between two servers 6
 - Using a single server..... 7
 - Rsam though a VPN, Proxy, or Citrix 7
 - Network performance and Rsam 7
- Database Considerations (on-prem customers) 8
 -  Updating DB Stats 8
 - Shared Database Systems 8
- Performance of Home Page Tabs 9
 -  It is Critical to optimize “common use” home pages 9
 - Using Widget “Priorities” 10
 -  Global Option “Auto-select first grouping....” 10
- Performance of Searches 13
 - Baseline 15
 - Cost of Sorting 15
 - Cost of Filtering 15
 - Advantage of Filtering on Record Type, Category Type, Object Type 16
 - Cost of Grouping 16
 - Indexing Record Attributes for Faster Filters, Sorting and Grouping 16
 - Cost of Permissions 17

Cost of Displaying Data	17
 Indexing Attributes	18
Best Candidates for Indexing	18
Implementing an Attribute Index	19
Indexing and large-scale data operations (imports and handlers)	20
Performance of Dashboards	21
Using Optimal Searches	21
Live Data vs Metrics Generator	21
Advantages in using Metrics for Dashboards	21
Disadvantages in using Metrics for Dashboards	22
Separate Complex Live Charts into Different Tabs	22
Performance of Record / Object Forms	23
Using Saved Search attributes in “Grid Mode”	23
Using on-screen handlers	25
Performance Affected by Permissions	26
Performance Affected by Custom Stored Procedures	27
Performance of Workflow Events	28
Performance of Import	30
Performance Troubleshooting Reference	33

About this Document

This document is intended to help Rsam administrators ensure they are getting optimal performance from their Rsam instances. This document focuses on key areas that every administrator should be aware of when building / updating their Rsam environment and configurations.

Typical Customer Performance

Good performance is one of the key elements of Rsam's development process. Rsam development and QA teams are constantly creating new and innovative methods to get the best performance from the application. Being a platform, however, the performance of a customer's Rsam instance is a shared responsibility between the Rsam application code, the customer's technical engineering, and the customer's design and configuration of Rsam.

While every customer shares the same Rsam application code, performance can vary from customer to customer depending on their environment as well as their Rsam configuration. The Rsam platform gives customers a great deal of power and flexibility, and the proper understanding and discipline around leveraging this flexibility is one of the major differentiators between customers with excellent performance, and those that struggle with performance.

Example - One customer can configure their "Submit Survey" button with one action, while another can configure it with 100 actions. With all other variables the same, the first customer will, of course, experience better performance than the second. Thus, the second customer should consider reducing the complexity of their configuration or introduce stronger hardware to compensate.

Most Common Causes of Performance Issues

While the current capacity and load placed on the system and network are critical elements to be aware of for good Rsam performance, a more common reason for performance issues is the over-loading and over-configuration of the pages, searches, forms, reports, workflow events & notifications configured in the Rsam instance.

Much of this guide will cover key areas to watch out for when configuring Rsam. It's very common for business users, who know about Rsam's flexibility, to request more and more information / charts / data elements / workflow logic, etc to be integrated into a single screen or event. While such items may be easy to configure, they all carry a cost in terms of performance and scalability. This document focuses on those costs so that customers can make informed decisions in their configuration.

While this document covers many guidelines for performance, we have marked some of the more key items with the following symbol ▲.

Cloud Customers vs On-Prem Customers

Some of the guidance in this document only applies to customers with Rsam installed on-premise in their own infrastructure. Cloud customers can ignore sections with the following label: **(on-prem customers)**

System Considerations (on-prem customers)

Systems should be sized, configured, and managed properly based on the customer's desired use of Rsam.

System Requirements

Rsam provides customers with a system requirements document, which provides general recommendations for system sizing based on the size of the use-cases. Customers should review this document and consider it to be the minimum adequate configuration. The items listed are good guidelines; however, there are far too many variables available to the Rsam administrator, to take them all into consideration.

For this reason, customers are recommended to either build the largest system they can reasonably build, or simply ensure that they have the ability and budget to expand their system's size in the future if needed.

Physical vs Virtual Machines

Rsam fully supports the use of Virtual Machines. In a perfect world Virtual Machines would perform just as well as systems with dedicated hardware. However virtual machines, by their nature, always have some degree of performance degradation. The degree of degradation varies on many factors, but it usually comes down to competing hardware and I/O resources with other operations that are running on the same hardware.

On average we see customers using Virtual Machines report about 25% less performance than those with comparable dedicated hardware. Sometimes this can be compensated for by increasing resources dedicated to the Rsam instance, however often there are I/O resources that cannot be managed by the virtual machine management software. And while 25% is a good average, this is completely dependent on the degree of competing resources. There have been some customers who experienced poor performance on their virtual machines only to find, later, that the system was completely overloaded with other applications running on other virtual machines (which are often hard for infrastructure teams to detect).

Dedicated physical hardware is always recommended and justifiable for customers with large instances of Rsam where consistent & predicable performance will be a key factor of success. Virtual machines are recommended for customers with fewer uses of the tool, and where consistent performance is not as critical.

Example - One customer was experiencing very slow performance from Rsam running on a Virtual machine. Several weeks were spent monitoring, reviewing, tweaking the system, but nothing helped improve the performance. It was noted that the same database / web server, when installed on another "lower-end physical system" performed very well. The Rsam instance was transferred to another virtual host with the same specifications, the same version, and in the same network... which ran about 300% faster. In the end it was discovered that the original host for the virtual machines was overloaded with other applications, despite configuring the virtual machine with its own dedicated CPU and RAM capacity.

Dedicated vs Shared Servers

Rsam supports installation on Shared Web servers and SQL server (servers that also host another application / database on the same instance). Such systems, however, have even more competition over resources than the virtual machines. When Rsam goes to execute a function on the Web or Database, it may have to wait for a function of another application to clear up or throttle down before the Rsam function can execute. This can cause performance delays.

Example - One customer reported poor performance with their database, but later found out that their database server was hosting databases for more than 80 other applications. Their database was moved to a dedicated database server (with far less CPU and RAM) and the customer saw their performance double and also found their performance to be far more consistent.

Importance of Fast Disks with SQL

The performance of Rsam greatly relies on the performance of the SQL database, and the performance of the SQL database is greatly dependent on the performance of the disk where that database resides. Each organization has its own standards for deploying SQL servers, but the following are recommendations for the SQL disk drive.

- Using a fast drive with high IOPS. High rotation speeds and SSD drives are recommendation.
- Using RAID drive configurations.
- Leveraging a local drive as opposed to a Storage Area Network. While some organizations have been able to achieve SAN speeds that match local drive speeds, performance is commonly much slower.

Separation of SQL Server and Web Server

The most common method to deploy Rsam (and other enterprise applications) is to place the SQL service and the Web service onto two separate servers and then have those two servers communicate with each other.

A commonly asked question is, "Would Rsam perform better if these two services are on the same system or on separate systems?" The answer depends on the scenario. In our exploration below, we will consider the following scenarios:

- Two servers.... Each with 2 processors and 16 GB of memory
- One server... with 4 processors and 32 GB of memory

Splitting between two servers

Using two servers (one for SQL and the other for Web) is a standard for many companies, and it works well with Rsam. It does cost more money, but it can perform better in some scenarios.

1. With two sets of hardware, resources outside of the CPU and RAM are doubled. For example: Disk, motherboard, network, and other I/O resources will be more or less loaded in this scenario.

2. With two sets of dedicated resources (including RAM and CPU), you can be certain that one service is never going to starve the other services by taking up all of its resources.
3. It is easier to monitor performance.

Using a single server

Using a single server for both SQL and Web can be very cost effective, and can also yield better performance in some scenarios, and can be especially useful in Dev and Test scenarios. Here are some points to consider:

1. Multiple processes and more memory on a single (larger) server typically cost less than having two smaller servers that are half the size.
2. Rsam's web and SQL server both take up memory and CPU when processing, but often it will be either the Web OR the SQL doing heavy processing while the other has less demand. In these scenarios, the services running will have more resources available since they can take advantage of the other services being more idle / less loaded.
3. Rsam's web server is constantly communicating with the database server in just about every interaction. Any network latency at all between the database server and the web server can cause performance issues. Having the web and the database on the same server eliminates the chance of communication latency.

Rsam through a VPN, Proxy, or Citrix

Many customers use Rsam through network tunnels or emulation environments, such as a VPN, Application Proxy, or a Citrix server. While these technologies work with Rsam and can perform well, each introduces an opportunity to introduce slower performance that is outside of Rsam's control. When using any of these technologies, make sure that attention is given to the fact that many low-volume network calls must be executed as the user navigates through Rsam. Delays introduced by a VPN, Proxy, or through Citrix can negatively affect user's performance.

Network performance and Rsam

Rsam pages are based on the Microsoft .Net framework which expects reasonable bandwidth in order to operate well. Slower networks or networks with a lot of contention can slow down the loading of pages for end-users. When using Ram over the Internet or at satellite locations, a good broadband connection is required for adequate performance. Slower DSL connections will work, but will be slow.

Database Considerations (on-prem customers)

Most Rsam functionality comes directly or indirectly from tables stored in the Rsam database. Good database performance is critical to the overall performance of Rsam.

▲ Updating DB Stats

Rsam recommends executing the SQL **SP_UPDATESTATS** procedure on the Rsam database using a daily scheduled job. Failure to do so can eventually lead to seemingly random performance impacts on query execution. **SP_UPDATESTATS** is a Microsoft procedure that performs background cleanup and realignment operations that can significantly affect performance of relational databases.

Shared Database Systems

Similar to the considerations of Virtual vs Physical machines is the consideration of having a dedicated vs shared Database server. Rsam makes heavy and continuous use of the database. While Rsam's database can reside on server that hosts databases for other systems, the database server should be adequately equipped to give Rsam significant database processing power. Any slow down because another application is making heavy use of the database services can have significant impact on Rsam.

Example - One customer reported that opening a record started taking more than 12 seconds (where previously it used to only take 2 seconds). After executing the **SP_UPDATESTATS** command, the customer was able to open the records in 2 seconds.

Performance of Home Page Tabs

During login and throughout general navigation in Rsam, users will visit one or more home page tabs many times. When users leave their home page screens to open a detailed view, they will ultimately return to the home screen to navigate to other objects or records.

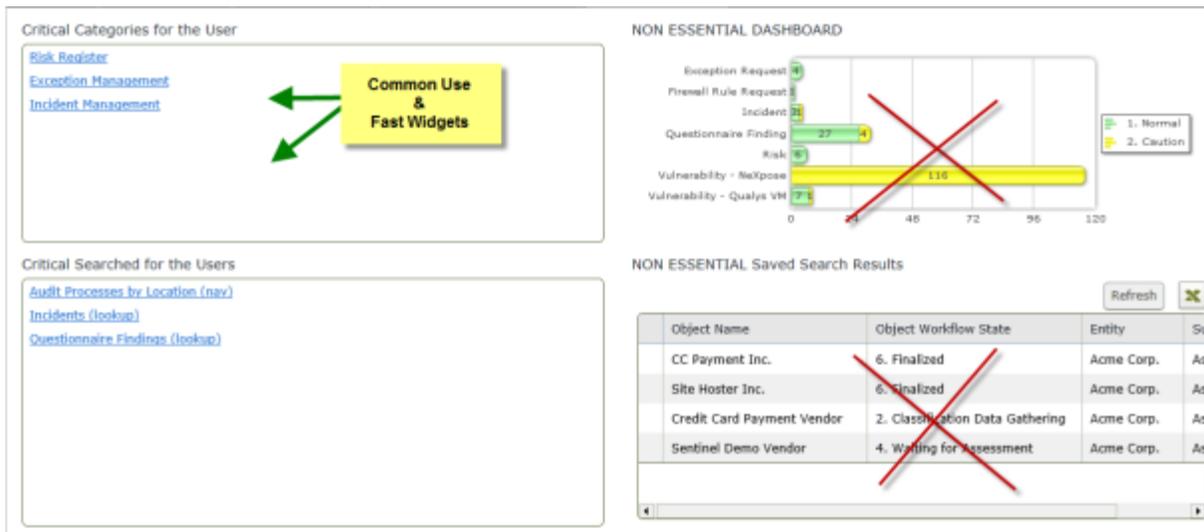
Proper consideration of performance in home page tabs is critical to maintaining a good user experience. This can be a challenge for the Rsam administrators as its common for users who understand the dynamic nature of Rsam to ask for more and more things to be placed into a single home page tab. However, if this is not managed, it can ultimately slow down their experience as well as the experience of others.

Since the home page tabs are visited frequently by each user in each session, understand that the impact of a home page that draws a lot of resources from the server will be magnified by the number of users accessing such tabs at any given time.

▲ It is Critical to optimize “common use” home pages

First and foremost, Rsam administrators should create quick and simple tabs that provide shortcuts to a user’s most common functions. For example, if your users continually need to access a specific set of searches or categories, make a home page tab dedicated to those items. AVOID placing Charts, Saved Search Result Widgets, and other heavy widgets on pages that users constantly use to perform their daily functions in Rsam. Otherwise the user will constantly have to wait and will constantly be demanding resources from the server to simply navigate back and forth between their records.

▲ While it is encouraged for customers to utilize many home page tabs, it is recommended to avoid adding higher cost widgets on the same tabs that provide common shortcuts. Consider placing charts, saved search results, reports and other widgets on their own tabs.



The screenshot shows a dashboard layout. On the left, under 'Critical Categories for the User', there are links for Risk Register, Exception Management, and Incident Management. A yellow box labeled 'Common Use & Fast Widgets' has arrows pointing to these links. Below this, under 'Critical Searched for the Users', there are links for Audit Processes by Location (nav), Incidents (lookup), and Questionnaire Findings (lookup). On the right, there are two widgets: 'NON ESSENTIAL DASHBOARD' which contains a horizontal bar chart with categories like Exception Request, Firewall Rule Request, Incident, Questionnaire Finding, Risk, Vulnerability - NoXpox, and Vulnerability - Qualys VM; and 'NON ESSENTIAL Saved Search Results' which contains a table with columns for Object Name, Object Workflow State, Entity, and Status. Both the chart and the table are crossed out with a large red 'X'.

Object Name	Object Workflow State	Entity	St
CC Payment Inc.	6. Finalized	Acme Corp.	As
Site Hostor Inc.	6. Finalized	Acme Corp.	As
Credit Card Payment Vendor	2. Classification Data Gathering	Acme Corp.	As
Sentinel Demo Vendor	4. Waiting for Assessment	Acme Corp.	As

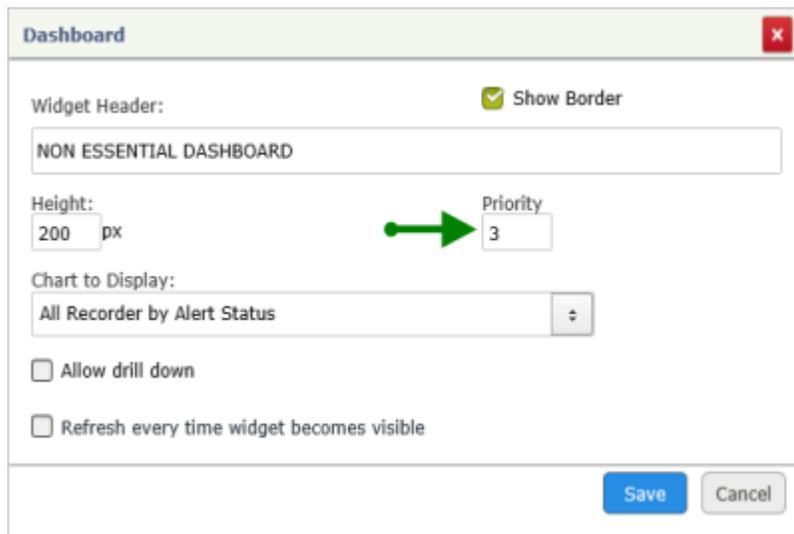
The following widgets tend to be the ones that require the most time and resources to load, and that should not be mixed with frequently used navigation tabs.

Widget Type	Resources Use
Saved Search Results Widget	Very High
SSRS Report Widget	Very High
Chart Widget with lots of data or multiple groupings	High

Example - One customer decided to enhance their main user page that contained shortcuts to generate new records, by adding in a few saved search result widgets and a couple of charts. While the screen was impressive, the screen that once took 1 second to load now took 6 seconds and users began to complain (since this was the screen they used continually throughout the day). The customer reverted the page to how it was originally and added a new tab next to it for users that wanted to see the other widgets.

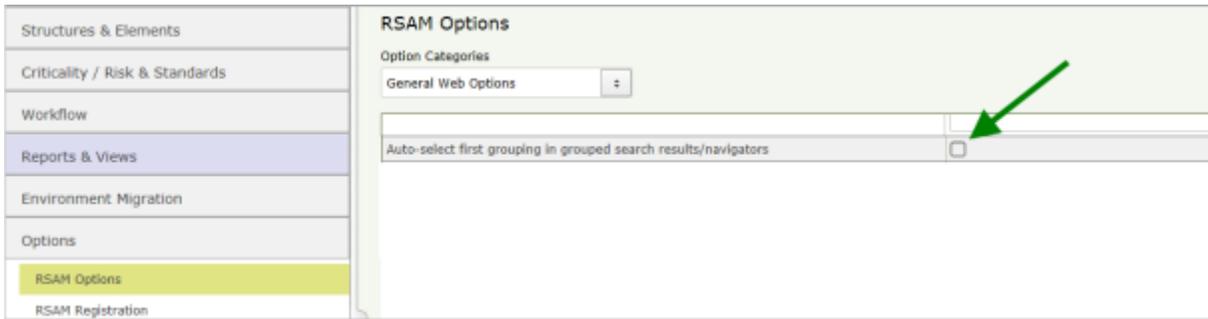
Using Widget "Priorities"

When clicking on a home page tab, all widgets (by default) begin to render simultaneously. However, in many tabs, there are certain widgets that are more commonly used than others (like category shortcut widgets vs chart widgets). In such a scenario, administrators can set the "priority" of the widget such that the important widgets start to render first, followed by other non-essential widgets. This allows users to access the priority widgets before the other widgets load.

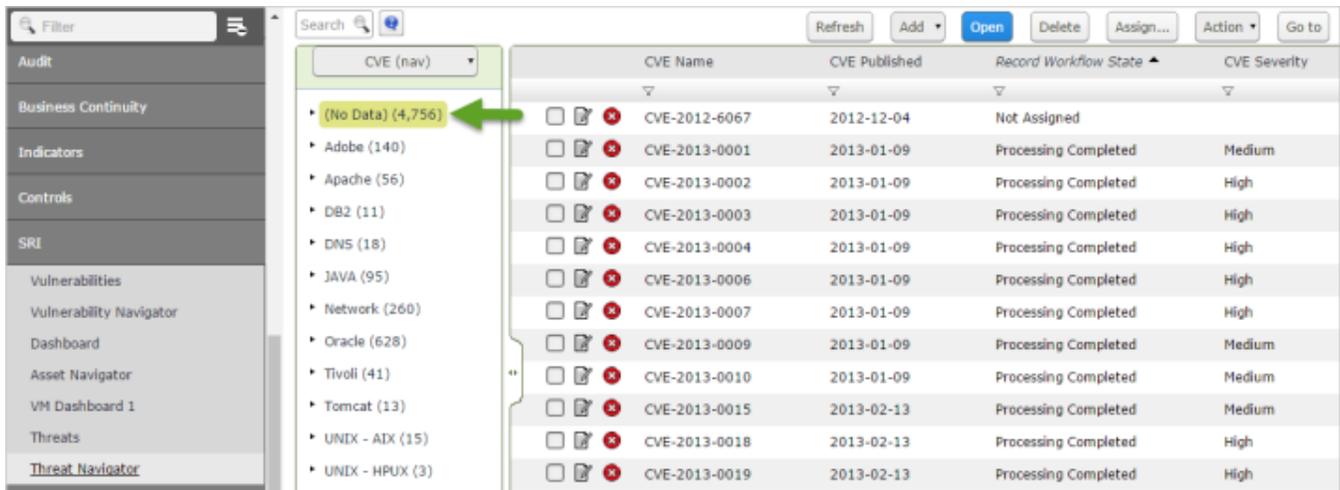


▲ Global Option "Auto-select first grouping...."

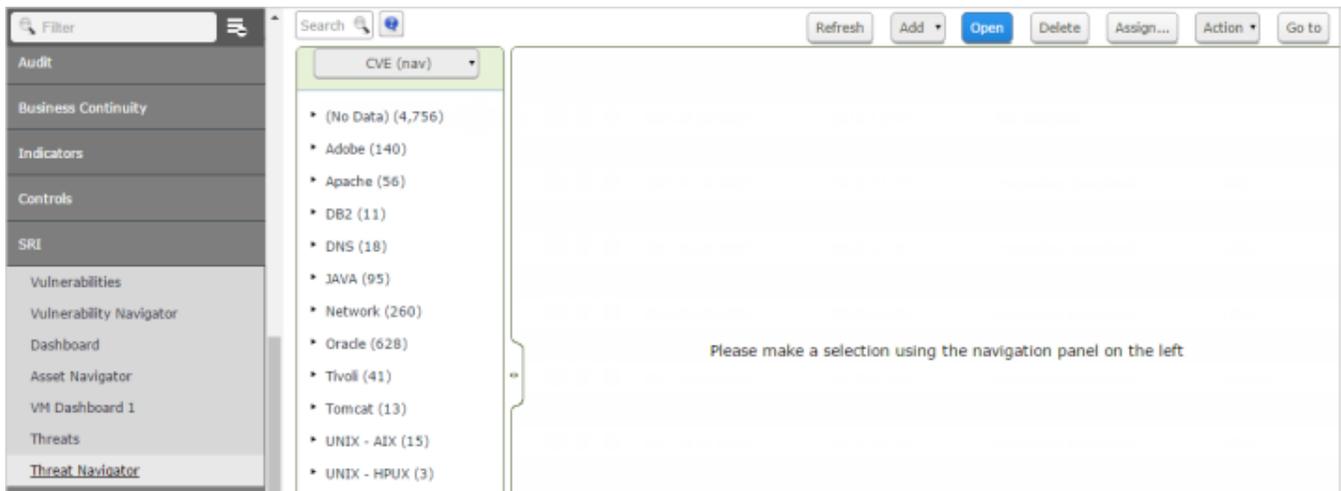
The global option of "Auto-select first grouping in grouped search results / navigators" can significantly impact the performance experience of users when utilizing home page navigators with large volumes of data.



If this option is enabled [X], then any navigator or search will auto-select the first grouped item in the left tree and display all search results related to that node in the right panel. In the following screen, we see that every time we enter this navigator it starts with the first node selected (which has several thousand records that the user may not care about).



The same screen with the option disabled only loads the grouping on the left, allowing the user to select the node before loading data. In this scenario, the screen loads much faster and produces far less load on the servers.



For optimal performance, it is highly recommended that administrators keep this option off.

Performance of Searches

Searches are one of the most common methods of using the data stored in Rsam. Performance of searches can affect not only users running the searches directly, but also the performance of dashboards, reports, home page widgets, home page navigators, lookup attributes and more.

Rsam's search code has undergone continuous optimization over the years, with a focus on returning large volumes of data quickly for the most common uses and scenarios.

Search performance is directly related to the features leveraged in the search criteria. Each feature has a performance cost associated with it, even when the feature is used to filter the overall search results.

Features that are targeted against values (like attributes, controls & criticality) have a higher cost than those that target object/record fields (like workflow state, record type, etc).

The items highlighted in Green below are examples of low-cost targets. Vs higher-cost targets that are highlighted in red.

Field Type	Field	Display Order
Object Fields	Object Name	1
Object Fields	Object Workflow State	2
Object Fields	Object Type	3
Object Fields	Object Type Model	4
Object Fields	Entity	5
Object Fields	Sub Entity	6
Record Fields	Record Category	7
Record Fields	Record Type	8
Record Fields	Record Workflow State	9
Record Attribute	CVE Name	10
Object Attribute	Threat Classification	11

The following is a summary of key search features that will affect performance followed by performance samples from a lab instance.

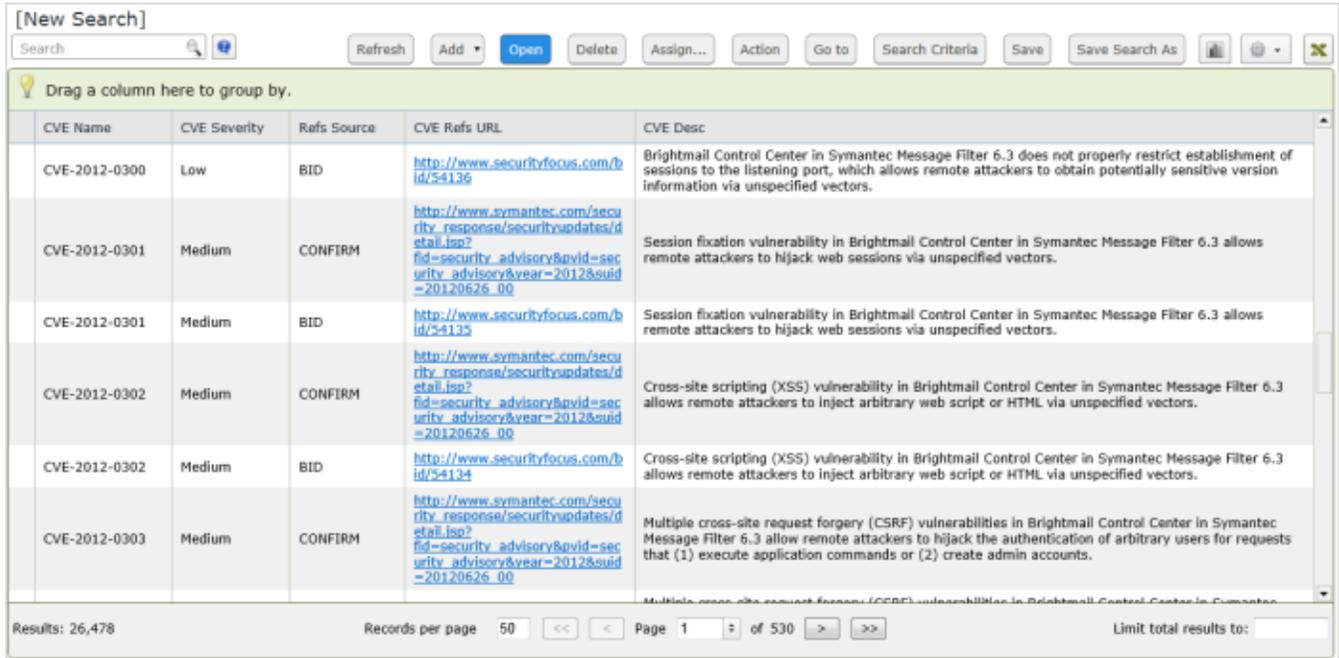
Search Feature	Target	Examples	Cost
Filter	<ul style="list-style-type: none"> Record Field Object Field 	<ul style="list-style-type: none"> Workflow State Record Type Category Type 	Low

Search Feature	Target	Examples	Cost
		<ul style="list-style-type: none"> Object Type 	
Filter	<ul style="list-style-type: none"> Unindexed Attribute Control Criticality Factor 	<ul style="list-style-type: none"> Severity Description Date 	Moderate
Filter with wild cards (like *VALUE*)	<ul style="list-style-type: none"> Unindexed Attribute Control Criticality Factor 	<ul style="list-style-type: none"> Severity Description Date 	High
Sort	<ul style="list-style-type: none"> Record Field Object Field 	<ul style="list-style-type: none"> Workflow State Record Type Category Type Object Type 	Low
Sort	<ul style="list-style-type: none"> Unindexed Attribute Control Criticality Factor 		Moderate
Group	<ul style="list-style-type: none"> Record Field Object Field 		Moderate
Group	<ul style="list-style-type: none"> Unindexed Attribute Control Criticality Factor 		High

▲ When making searches, you should only apply filters, sorting, and grouping where necessary. Each filter, sort, and group adds to the time and processing power required to execute the search.

Baseline

The following examples are drawn from an instance of Rsam with a million records. This search is against a record type that has 26,000 records with no sorting, filtering, or grouping rendered in less than **2 seconds**.



CVE Name	CVE Severity	Refs Source	CVE Refs URL	CVE Desc
CVE-2012-0300	Low	BID	http://www.securityfocus.com/bid/54136	Brightmail Control Center in Symantec Message Filter 6.3 does not properly restrict establishment of sessions to the listening port, which allows remote attackers to obtain potentially sensitive version information via unspecified vectors.
CVE-2012-0301	Medium	CONFIRM	http://www.symantec.com/security_response/securityupdates/detail.jsp?fid=security_advisory&year=2012&suid=20120626_00	Session fixation vulnerability in Brightmail Control Center in Symantec Message Filter 6.3 allows remote attackers to hijack web sessions via unspecified vectors.
CVE-2012-0301	Medium	BID	http://www.securityfocus.com/bid/54135	Session fixation vulnerability in Brightmail Control Center in Symantec Message Filter 6.3 allows remote attackers to hijack web sessions via unspecified vectors.
CVE-2012-0302	Medium	CONFIRM	http://www.symantec.com/security_response/securityupdates/detail.jsp?fid=security_advisory&year=2012&suid=20120626_00	Cross-site scripting (XSS) vulnerability in Brightmail Control Center in Symantec Message Filter 6.3 allows remote attackers to inject arbitrary web script or HTML via unspecified vectors.
CVE-2012-0302	Medium	BID	http://www.securityfocus.com/bid/54134	Cross-site scripting (XSS) vulnerability in Brightmail Control Center in Symantec Message Filter 6.3 allows remote attackers to inject arbitrary web script or HTML via unspecified vectors.
CVE-2012-0303	Medium	CONFIRM	http://www.symantec.com/security_response/securityupdates/detail.jsp?fid=security_advisory&year=2012&suid=20120626_00	Multiple cross-site request forgery (CSRF) vulnerabilities in Brightmail Control Center in Symantec Message Filter 6.3 allow remote attackers to hijack the authentication of arbitrary users for requests that (1) execute application commands or (2) create admin accounts.

Results: 26,478 Records per page: 50 Page 1 of 530 Limit total results to:

Cost of Sorting

Adding one or more **sorts** increases the cost of the results. Sorts by low cost targets are faster than sorts against high cost targets

Search Function	Records	User	Seconds
Unsorted & Unfiltered	26,000	User 1	1.5
Sorted by 1 "low cost" target	26,000	User 1	2
Sorted by 1 "high cost" target	26,000	User 1	4

Cost of Filtering

Adding one or more **filters** increases the cost of the results. Filters in low cost targets are faster than Filters in high cost targets. Filters also vary based on data size and the use of wild cards like "*".

Search Function	Records	User	Seconds
Unsorted & Unfiltered	26,000	User 1	1.5

Search Function	Records	User	Seconds
Filtered by 1 "low cost" target	26,000 (filtered to 15,000 results)	User 1	2
Filtered by 1 "high cost" target	26,000 (filtered to 15,000 results)	User 1	3
Filtered by 1 "high cost" target using wild card *	26,000 (filtered to 15,000 results)	User 1	4

Advantage of Filtering on Record Type, Category Type, Object Type

While filtering can have a cost, the basic properties (like Record Type, Category Type & Object Type) have a very small cost, and they are applied first to the filtering process. If your search targets a specific category type / record type it's highly recommended that you include them in your search filters.

Cost of Grouping

Adding one or more **groupings** significantly increases the cost of the results. Grouping by low cost targets is faster than grouping by high cost targets. Also, the number of resulting groups (displayed on the left) significantly impacts the performance.

Search Function	Records	User	Seconds
Unsorted & Unfiltered	26,000	User 1	1.5
Grouped by 1 "low cost" target (3 groups)	26,000	User 1	3
Grouped by 1 "high cost" target (3 groups)	26,000	User 1	5
Grouped by 1 "high cost" target (1,000 groupings)	26,000	User 1	30

Indexing Record Attributes for Faster Filters, Sorting and Grouping

The performance cost of filtering, sorting, and grouping on attributes assumes an "Attribute Index" has not been applied. We will cover the major performance benefit from record attribute indexes in the next section.

Cost of Permissions

Distribution of permissions has a significant impact on search performance. Permissions assigned at a higher level (like an entity or sub-entity) require very little processing to parse and are very quick to calculate. Permissions assigned at a detail level (like an individual record) require more records to parse.

How permission assignments affect performance has less to do with “how many things can I access” and more to do with “how many items have I been assigned permissions to directly”.

Example

- A user with access to 100,000 records, but was assigned just one time to the overall entity has a very low performance cost
- A user with access to 100,000 records, and they were assigned to each of those records individually has a very high-performance cost

The following users were utilized in our performance examples:

- User 1 = Administrator
- User 2 = Access to all records in the object (only assigned to the object)
- User 3 = Assigned individually to all 26,000 records (assigned to all records individually)

Search Function	Records	User	Seconds
Flat Unsorted & Unfiltered	26,000	User 1	1.5
	26,000	User 2	3
	26,000	User 3	20

Cost of Displaying Data

Note that there is very little cost to showing multiple columns of data on the screen. While filtering, sorting, and grouping on attributes (for example) require more search execution time, customers can “display” as many attributes as desired without great concern for performance.

▲ Indexing Attributes

It is often desirable to perform record searches and create charts and navigators based on attributes, despite the higher cost in performance. This is where Attribute Indexing comes into play.

Customers can select specific attribute types that they wish to “index” for a performance boost. The impact of proper indexing can be significant. For example, using a basic SQL server, returning 50,000 rows grouped may take 5-10 seconds without a proper index. The same operation may only take 1-2 seconds with valid index in place.

For most customers, the record volume is 20x -50x that of their object volume.

While indexing significantly boosts performance when reading data, it does slow down the writing of the data. This negative performance impact will be mostly visible during large-scale data import operations and large-scale handler execution.

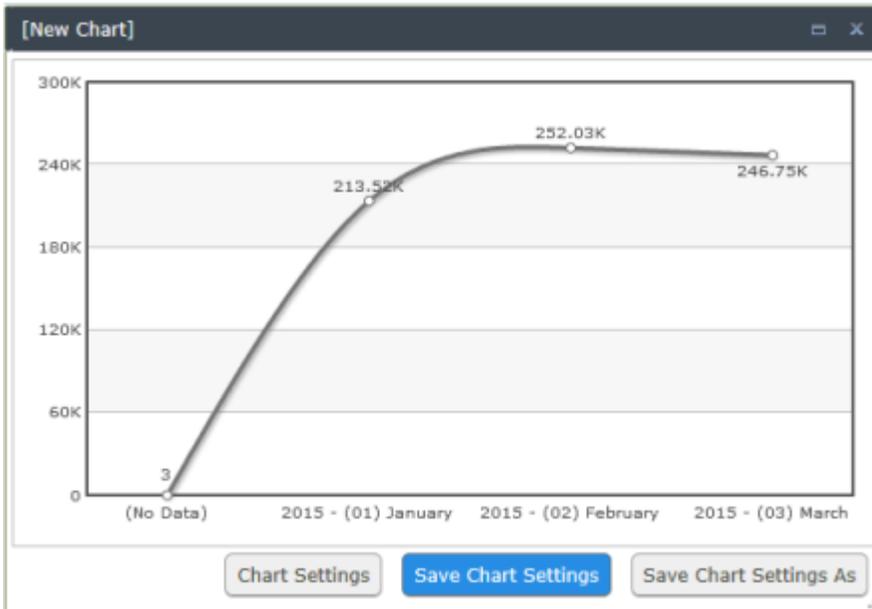
Best Candidates for Indexing

The following attributes are excellent candidates for indexing:

- 1. Any attributes that are used in common navigator **groupings**

▼ Open (712,349)	Host - 11122	FCBB831F 16D5	5967DAB1 C005 49C8 A010 A8967	59
(No Data) (2)	Host - 21664	1EFA3CF3 538A 4	2DBD55AD	95
1. Very Low (142,467)	Host - 21764	1D980	A3A7EA4A 9287 4DA8 A6	46
2. Low (142,473)	Host - 18842	E83DF273 C223	4EAA9374 DCBF 421C 9412 C456C	87
3. Medium (142,472) ←	Host - 13712	10BDD47D 0F7E	DD5C25D3 8914 4427 8DEF 10BDD	63
4. High (142,468)	Host - 14012	4213F164 B647	3188990C F879 4C90 90E6 513A0	81
5. Critical (142,467)	Host - 14112	76D115AE E371	464F9AF5 3D0F 4B19 A67A FECC3	32

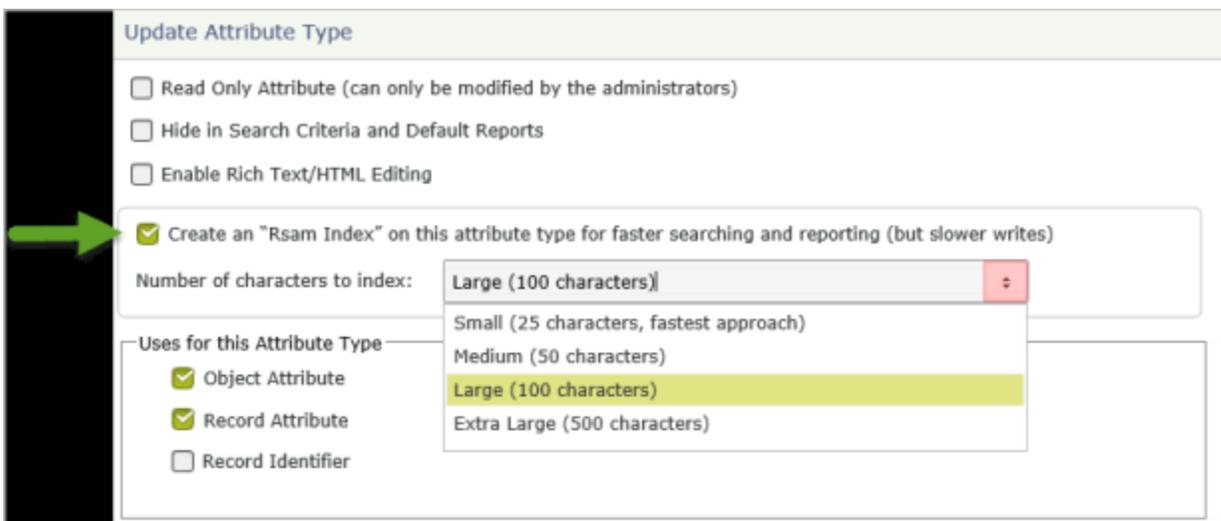
- 2. Any attributes that are commonly used in chart groupings



3. Universal attributes that are commonly used in all records:
 - a. Universal Severity
 - b. Date Opened
 - c. Etc

Implementing an Attribute Index

The Rsam administrator guide / online covers how to build attribute indexes. In general, this is a simple selection that is made in the attribute type admin screen. However, the administrator guide / online help review a variety of considerations / options when leveraging attribute indexes.



Update Attribute Type

- Read Only Attribute (can only be modified by the administrators)
- Hide in Search Criteria and Default Reports
- Enable Rich Text/HTML Editing
- Create an "Rsam Index" on this attribute type for faster searching and reporting (but slower writes)

Number of characters to index: **Large (100 characters)**

- Small (25 characters, fastest approach)
- Medium (50 characters)
- Large (100 characters)**
- Extra Large (500 characters)

Uses for this Attribute Type

- Object Attribute
- Record Attribute
- Record Identifier

Indexing and large-scale data operations (imports and handlers)

While the attribute indexing can have a dramatic effect on areas where Rsam reads those attributes from the database, it can also have a negative effect on performance when writing to the database. Writing the attributes to the database will take anywhere between 10% - 40% longer depending on the size of the index.

Performance of Dashboards

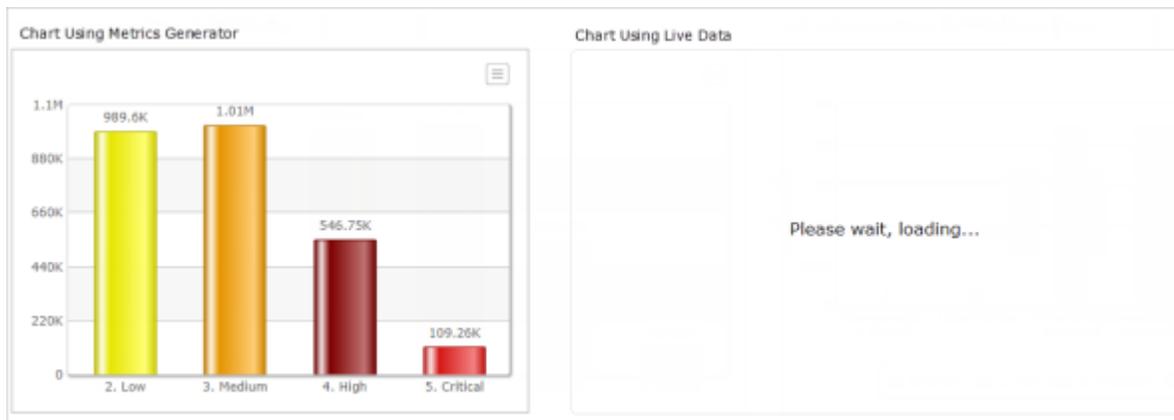
Customers make extensive use of Rsam's charting feature to present dashboards to their end users and executive management. Performance of dashboards is important for a good customer experience.

Using Optimal Searches

Rsam charts leverage saved searches for their data set. The vast majority of the time that it takes to show a chart comes from the performance of the search. Optimizing searches given the recommendations above is key to the performance of a chart.

Live Data vs Metrics Generator

Rsam charting can be used to represent very large sets of data based on Search data. The larger and more complex the search and data sets are, the longer it will take for the chart to generate. Customers wishing to display larger data sets will gain significant performance benefits using the Metrics Generator Feature.



Metrics Generator is a feature of Rsam that allows organizations to execute a search during off-hours, preprocessing the statistics that are interesting for a chart. These stats are written back into a smaller, more concise set of records. Charts generated from pre-processed metrics are exponentially faster and require minimal resources to generate.

Advantages in using Metrics for Dashboards

- Much faster display of large datasets to a user
- Better usage of System resources (to avoid slowing down other users)

Disadvantages in using Metrics for Dashboards

- Data displayed is not live data, but rather data that existed when the metrics generator was processed.
- Users can only drill down to the metric data, not the individual records that generated the metric
- Administrators must plan for what metrics to generate and perform the necessary configuration.

Separate Complex Live Charts into Different Tabs

When it is important to display live charts with large or complex datasets, it is beneficial to separate these charges onto different tabs. The number of charts that can be displayed quickly varies depending on the data size and complexity.



Performance of Record / Object Forms

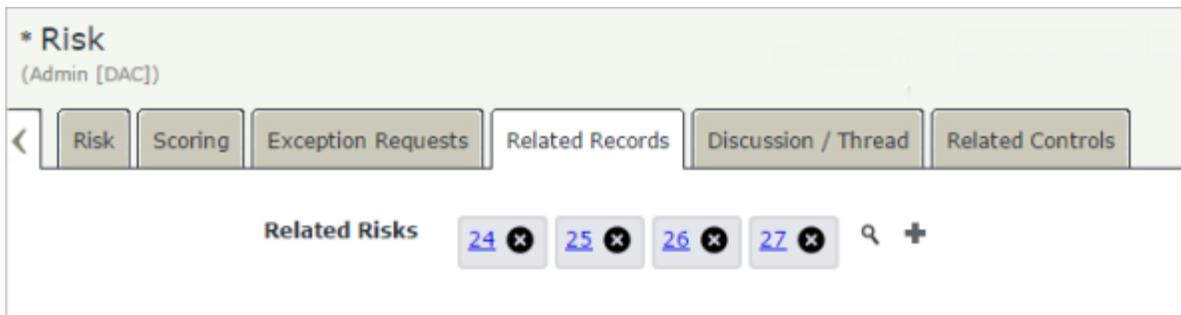
Record and Object forms are where most end-users spend their time entering and reviewing data. It is very important to keep these forms quick & responsive. Like most other areas of Rsam, the configuration of these forms and what is shown in each tab has a major effect on performance. While these forms tend to load quickly and efficiently, there are a few features that can detract from the performance.

Using Saved Search attributes in “Grid Mode”

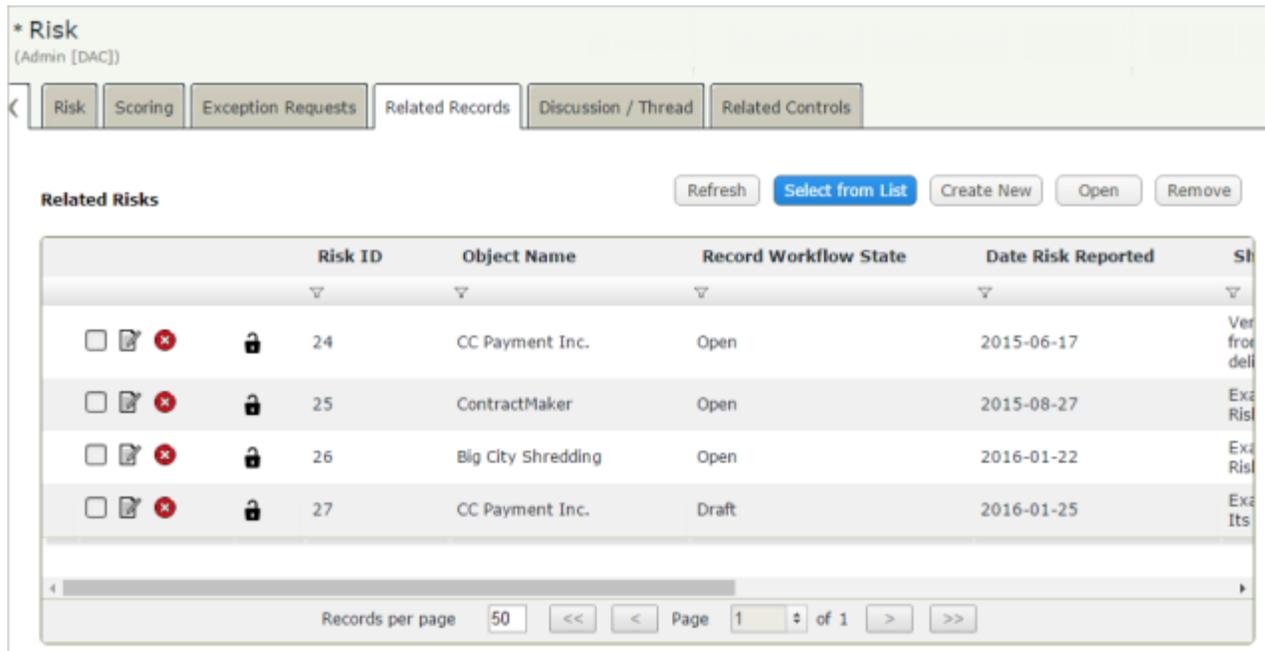
Saved search attributes have a powerful feature called the “Grid Mode”. This is where a user is shown not only a link to the related records, but also presented with several columns of data from the search that is associated with the attribute.

While this mode can be very helpful for the user, it does come at a considerable processing cost. Rsam must not only load the selected values, but also execute the search and load the appropriate columns for each selection.

Example - “Simple Mode”



Example - "Grid Mode"



While using grid mode is encouraged, follow these simple guidelines:

1. **Use Simple mode where it makes sense:** In scenarios where grid mode is not important, be sure to use simple mode. Simple mode is extremely fast (instantaneous). The user can still click the link to access the related item to see its details, thus the main drawback is simply that you will not see multiple columns of data immediately.
2. **▲ Avoid putting a grid on the first tab:** Since the grids can take time to load, avoid putting them in the first tab of a record. Otherwise loading the record will take longer.
3. **▲ Place grids in their own tab:** Grids are loaded "on-demand", by placing the grid into a separate tab, the user will not have to wait for the grid to load unless they click on the specific tab where the grid exists. Allow users to quickly open and edit a record by moving the Grid attributes outside of the main data entry tabs. Also, avoid putting too many grids on the same tab for the same reason.

Consider the following performance samples:

Function	Details	Seconds
Opening a Record	20 attributes and no Grid	2
	20 attributes and 1 Grid with 5 columns	5
	20 attributes and 2 Grids with 5 columns each	9
	20 attributes and 2 Grids (but grids are not on the first tab)	2

Example - One customer placed three grid-mode attributes on the initial tab for one of their main record types. The tab was taking 6 seconds to load. After moving these grids to another tab, the record loaded in 1-2 seconds.

Using on-screen handlers

On-Screen handlers are a great feature of Rsam allowing you to dynamically adjust the look and contents of a form based on the user's interactions. Rsam does not restrict any number of handlers from being associated with a single form. When loading a record or object, however, all on-screen handlers have to be loaded and all rules begin to monitor user interactions. Loading large numbers of handlers can cause smaller delays in loading / navigating through a form.

Performance Affected by Permissions

Rsam's permission capabilities allow customers to assign permissions at just about any level in the system including at the entity, sub-entity, object, record parent and record children levels. As users login to Rsam, open forms, execute searches, view charts and reports, and more, their actions are constantly referenced back to their permission assignments. Permission assignments can have a significant impact on overall performance for Rsam.

 As a general guideline, try to **assign permissions at the highest possible level**. Assigning permissions to individual records is fine and sometimes necessary; however, this does come with a cost when dealing with high volumes of records (tens-of-thousands to millions).

Consider the following scenario:

1. The system contains 100 objects, each with 1,000 records (100,000 records in total).
2. User A is assigned a role at the entity level that contains all objects. In processing permissions, Rsam must only parse through a single permission record. Performance will be good.
3. User B is assigned a role to each object individually. In processing permissions, Rsam will have to parse through 100 individual permission records. Performance will be good here as well.
4. User C is assigned a role to each record individually (likely due to some automated assignment process). Rsam will have to parse through 100,000 permission records to determine the user's proper access. Performance will be slower for this user.

Note: It is very common to assign permissions on a record-by-record basis. However, for higher volume use-cases, assigning individual permissions to so many individual records will affect performance.

Performance Affected by Custom Stored Procedures

Rsam allows customers to execute custom SQL stored procedures to perform custom actions with records & objects. Executing a custom procedure can be instantaneous, or take a significant amount of time depending on the nature of the procedure and what it's executing in the database.

Customers that have performance concerns should check to see if that area of the process is executing a custom stored procedure. If so, the procedure should be reviewed (using the SQL interface) to determine how much time it's taking to execute that procedure.

Places where customers might have leveraged custom stored procedures include

1. Workflow Button Clicks (directly or via handlers)
2. Adding a Record or Object (directly or via handlers)
3. Importing a Record of Object (directly or via handlers)
4. Loading a custom home page widget

Note: Custom stored procedures written by the Customer or Rsam staff are not a part of standard Rsam technical support. If you need assistance with a custom stored procedure, please contact your Rsam sales representative.

Performance of Workflow Events

Rsam's dynamic workflow capabilities provide administrators extensive functional capabilities. Rsam imposes no restriction on the number of events that can occur during any workflow event so it is important that customers temper their desire for large / complex operations with the performance for their end-users.

1. **Avoid Excessive use of handlers on a single button click:** While customers are encouraged to execute as many handlers as necessary to enable their workflow, keep in mind that each handler execution has a performance cost. Excessive handlers that are executed during a workflow operation (like a button click) can have performance impacts.
 - a. Eliminate any unnecessary handlers
 - b. Offload large handlers to nightly scheduled tasks where applicable
2. **Avoid using lots of handlers on the Save / Update button:** The save & update buttons are workflow button clicks as well. Customers have the opportunity to add handlers to these buttons to perform special operations whenever a user saves their record. However, any delays when clicking on a common button like "save" can be frustrating to the user.
 - a. Avoid using handlers in the Save buttons, if possible
 - b. If handlers are required for some record types and not others, make dedicated "save" buttons specific to those record types, leaving the general / default "save" buttons clean.
3. **▲ Avoid using lots of handlers on the "on-create" feature:** Rsam allows customers to execute handlers on the "Create" event for an object or record. However, these handlers execute (and can impact performance) during the creation of all records / objects. Even if criteria are used to isolate handlers to a specific type of object or records, Rsam must first evaluate the handlers during record creation.
 - a. Consider placing the execution of handlers on the first submit button executed for the record instead of the on-create.
4. **▲ Using saved searches to provide a target selection in scheduled events:** When creating scheduled events, during target selection, selecting items using saved searches will improve the performance drastically. The following image shows the target selection options.

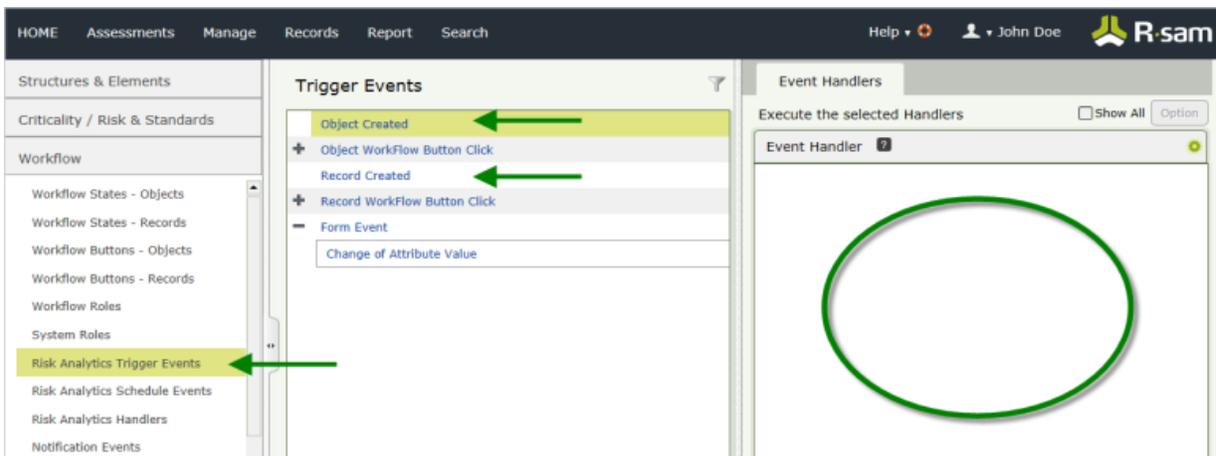
Event Name:
SOAR TM Related Platforms SP Schedule Daily
Event Description:
Target Selection
<input type="radio"/> Manually select target(s)
<input type="radio"/> Target items using Record Saved Search(s)
<input type="radio"/> Target items using Object Saved Search(s)
Target: execute this event on the following target(s)
Threat and Vulnerability Management

Note: Ensure the handlers do not have repetitive criteria when using Saved Searches for target selection.

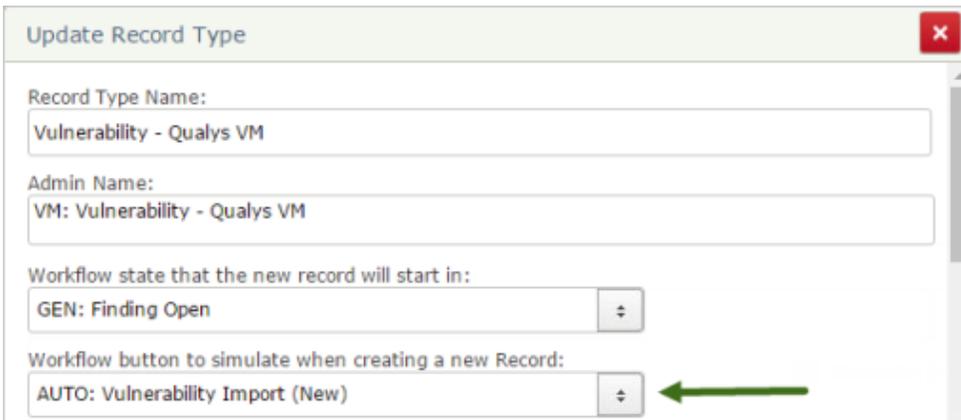
Performance of Import

Many customers import high volumes of information on a regular basis into Rsam. Rsam’s native import operations execute very quickly and efficiently. A basic import function that is not weighed down by any extra workflow or logging operations can import tens of thousands of records in a couple of minutes. However, the way import and handlers are configured can greatly affect the performance of import operations.

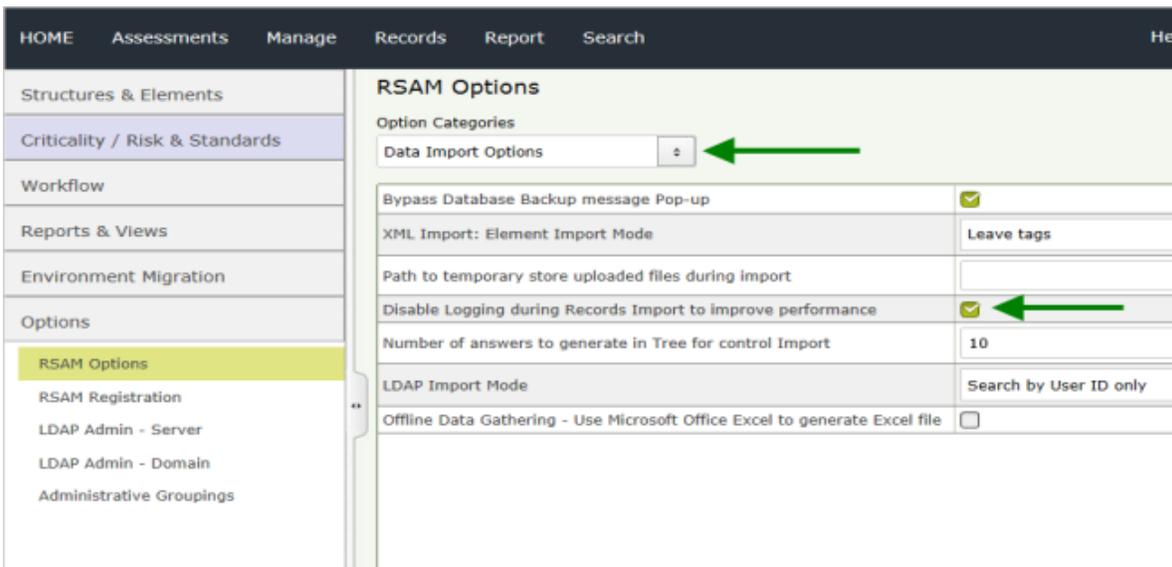
- ▲ Cost of Actions that execute Handlers:** Handlers are extremely powerful, but can be very costly during an import operation. With the dynamic capabilities of handlers, Rsam must evaluate the handler criteria and execute the handler actions for each record being processed. This can quickly turn a 10-minute import into a 60-minute import. Keep handlers that execute during import to a minimum. Some handlers will be required to meet the needs of the organization, but often time’s customers execute additional handlers that have minimal benefit, not realizing the cost. Reducing the handlers executed during an import can greatly improve the speed of the import.
- ▲ Handlers used on “Created” trigger events (Records or Object):** This is important enough to warrant its own section. When looking at import performance, you must also check your Object / Record created handlers that have been configured. Even if the handlers are not intended for the objects / records you are importing, they will still be evaluated to determine if they apply to each record. It is highly recommended that customers executing larger imports or other large-scale operations minimize or eliminate the use of “Object Created” and “Record Created” handlers and, instead, incorporate those handlers on button clicks.



Instead of using the event on-create feature, customers should leverage the “simulate workflow button” feature inside of each record type / object type admin screen. This provides the same functionality, but more targeted to that type of record or object.



3. **Logging of Import Activities:** Rsam includes an option to disable logging of data changes during an import. Since logging such operations requires Rsam to write a log record for every item created and for every value populated, disabling logging during import can improve the speed by as much as 50%.



4. **▲ Importing against LDAP Attributes:** Often times we need to import into an LDAP Lookup attribute (an attribute that references a user in the corporate directory). If, however, the data source references an account that is new to Rsam, Rsam must:
 - a. Query the LDAP server (over the network)
 - b. If found, create the user account
 - c. Make any necessary assignments

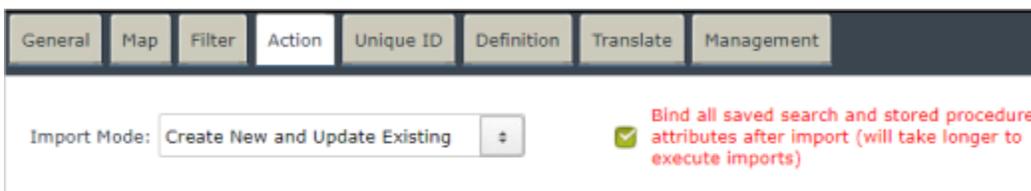
These operations (especially the LDAP Query) can be costly when it comes to import performance. If you are importing from a data source that references user accounts that do not yet exist in Rsam, allot extra time for these imports to execute.

- 5. **Importing Invalid Records:** Imports that contain a large number of “invalid values” can also affect performance. To detect an invalid value (such as a listbox value that is invalid or an LDAP user that does not exist in the LDAP Domain) takes longer than importing a value that is valid. Both the SQL database and LDAP services have to look through all existing values to determine something is invalid.
- 6. **Number of Identifiers:** When opening in create & update mode, Rsam uses “identifiers” to see if the record being imported already exists. The more identifiers that are used the slower the performance will be for this operation. Having more than 2 or 3 identifiers per record type can significantly impact performance.

Examples of Import Timing

Objects Created	Records Created	Attributes Mapped	Handlers Involved	Minutes to Execute
100	10,000	10 (Text, Listbox, Date, Numeric) with 1 identifier	0	2.5
100	10,000	10 (Text, Listbox, Date, Numeric) with 1 identifier	1 on Button Click	3.25
100	10,000	10 (Text, Listbox, Date, Numeric) with 1 identifier	10 on Create	10.5
100	10,000	10 (Text, Listbox, Date, Numeric) with 5 identifiers	0	10

- 7. **Impact of binding saved search and stored procedure attributes after import:** During object / record import, when mapping the import profile, in the **Action** tab, if the check box to bind saved search and stored procedure attributes is selected, then it takes a longer time to execute the import.



Performance Troubleshooting Reference

If you feel you are having poor performance, you can use the following list to check for some of the most common performance culprits.

Performance Issue	What to Check
All Issues	<ul style="list-style-type: none"> • Are the servers adequately sized? • Are the virtual / shared servers competing with other high-demand applications? • Is there latency on the network? • Is your SQL server configured to run SP_UPDATESTATS on a Daily Basis? • Do you have custom stored procedures that are executing in Rsam that may take extra processing? • Have you tried the same operation on another server to see the performance difference? • Have you tried using another browser / version? • Is Rsam extended logging / debug enabled?
Slow Login	<ul style="list-style-type: none"> • Is the LDAP Server / SSO system performing well? • Does the user have excessive individual permissions assigned to individual records or objects? • Is the initial home page loaded with lots of high-cost widgets?
Slow Loading a Home Page Tab	<ul style="list-style-type: none"> • Is the initial home page loaded with lots of high-cost widgets like Saved Search Results, SSRS Reports, or Big Charts? • Does the user have excessive individual permissions assigned to individual records or objects? • Are you using a custom stored procedure for one or more of the widgets?
Slow Loading a Search	<ul style="list-style-type: none"> • Are there many high-cost search operations (multiple sorts, filters, grouping by attributes)? • Have you indexed attributes that are used for sorting, filtering or grouping your search? • Are there a large number (50+) of groups in the search results?

Performance Issue	What to Check
	<ul style="list-style-type: none"> • Does the search utilize wild cards in large volume text attributes? • Does the user have excessive individual permissions assigned to individual records or objects? • Are you using a custom stored procedure for the search?
Slow Loading a Record	<ul style="list-style-type: none"> • Does the record's tab have saved search attributes in grid mode? • Does the record have a large volume of on-screen handlers?
Slow Importing Data	<ul style="list-style-type: none"> • Are multiple handlers associated with the on-create action (in the workflow screen, not the import screen) • Are multiple handlers being executed on a button click designated in the action tab? • Are many identifiers (more than 3) being used? • Are you importing LDAP attributes for users that do not currently exist in Rsam? • Have you tried disabling the import logging from the Options screen? • Are you executing a custom stored procedure during any of the actions during the import?
Slow Clicking a Workflow Button	<ul style="list-style-type: none"> • Are many multiple handlers executing on the button click, or handlers that have large scope? • Are you executing a custom stored procedure on the button click?