

IBM® IoT

Maximo Asset Management –7.6 Release

Maximo Report Performance

Pam Denny

Maximo Report Designer/Architect



CONTENTS

- Revision History iv

- 1 Configuration 8
 - 1.1 Configuration: Clustering9
 - 1.2 Configuration: BIRT Report Only Server..... 12
 - 1.2.1 Configuration Recommendations and Notes: 19
 - 1.3 Configuration: Replicated, Reporting Databases.....20
 - 1.4 Configuration: Hardware21

- 2 Designing 23
 - 2.1 Design: Analysis Options24
 - 2.2 Design: Format Impacts25

- 3 Developing 29

- 4 Administration 30
 - 4.1 Administration: Record Limits31
 - 4.2 Security Group Limits: Server Limits..... 33
 - 4.3 Security Group Limits: Preview Limits.....34
 - 4.4 Security Group Limits: Scheduled Report Limits36
 - 4.5 Minimum versus Maximum Value of Security Group limits37

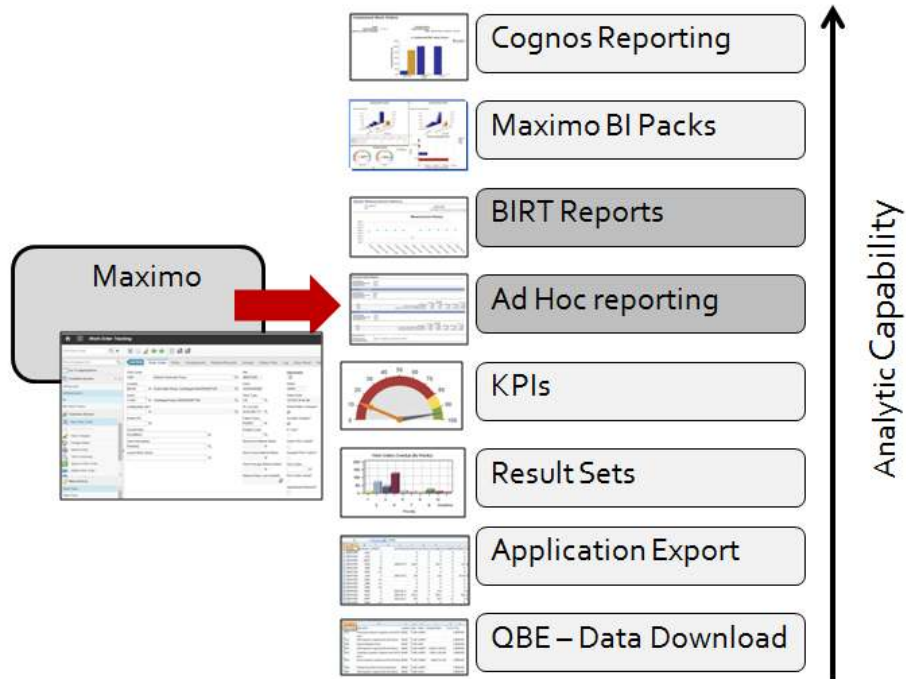
4.6	Administration: Queuing	38
4.7	Administration: Performance Tab	40
4.8	Administration: View Report Processing	41
4.8.1	Administration: Report Cancellation.....	42
4.8.2	Administration: Managing Report Cancellation	43
4.9	Administration: Report Escalations	44
5	Execution	45
5.1	Execution - Scheduling.....	46
5.2	Execution: Schedule Only Reports.....	47
5.3	Execution - Schedule Only Reports – Reserved Processing Time	48
5.4	Execution - Report Usage	50
6	Additional References	51

REVISION HISTORY

Date	Version	Revised By	Comments
November 2015		PDenny	Correction on page 13 to BROS
December 2014		PDenny	Initial Release

Overview

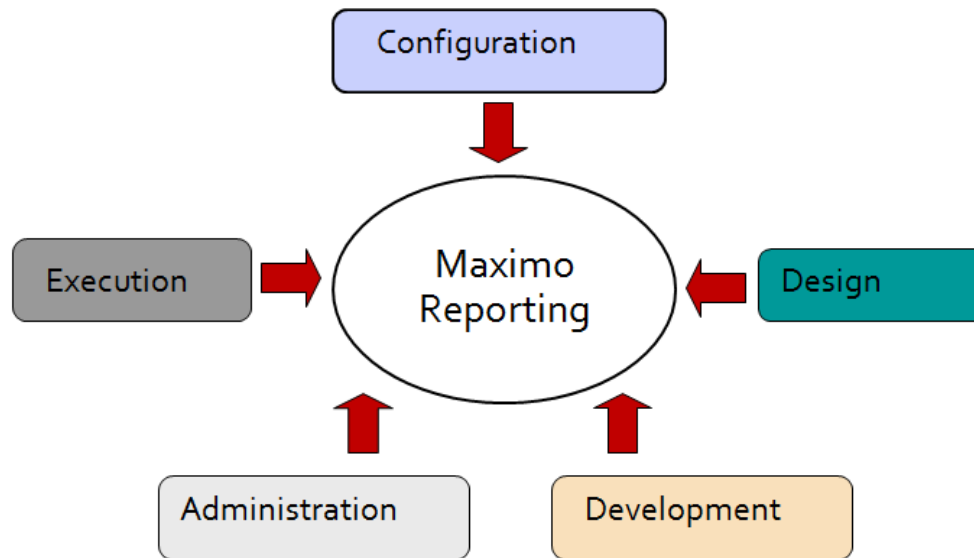
To respond to today's dynamic Business Environment critical business information needs to be analyzed immediately. Maximo's Business Intelligence (BI) tools provide a variety of ways in which you can view, analyze and take action on all the detailed information which is collected in the various Maximo applications. These suite of tools are shown below in hierarchy order of their increasing analytic capability and include application exporting, result sets, ad hoc reporting, along with BIRT and Cognos reporting.



To maximize your use of these tools, this document focuses on the performance features of two of the BI tools highlighted above - Ad Hoc reporting and BIRT reporting. These tools utilize the BIRT report engine, and a variety of components can impact their report performance.

To highlight this, the guide will review the five major reporting processes: Configuration, Designing, Developing, Administering and Execution. As a best practice, you should review each of these processes to maximize report performance in your environment. If any of the five components is not analyzed, your entire reporting process will not be optimized.

*Note: This document applies to the Maximo® Version 7.6 Release with BIRT Reporting only.



Configuration

Configuration involves not only the architecture of the entire System, but also its hardware specifications (CPU, Disk Space), Network Connections and Directory Service (LDAP). This document will focus on the areas directly impacted by performance including: Clustering, Report Only Server, and Replicated Databases .

Designing

The design of a report greatly impacts a reports performance. A simple list report with a single grouping could take just a few seconds to execute, whereas a complex analysis report containing multiple subreports, graphs and groupings could take much, much longer.

Additionally, the best tool for the business requirement should be reviewed. For example, users may only need to know the status of a particular business event, like 'Number of Work Orders Outstanding' and metrics like this may be better created and maintained via a KPI than a report.

Developing

The development of a report is very critical due to its sql statement. An optimized sql statement enables a report to execute at its peak. However, using non-optimized sql statements with incorrect table joins can cause a report to not only return incorrect data, but execute inefficiently consuming critical system resources.

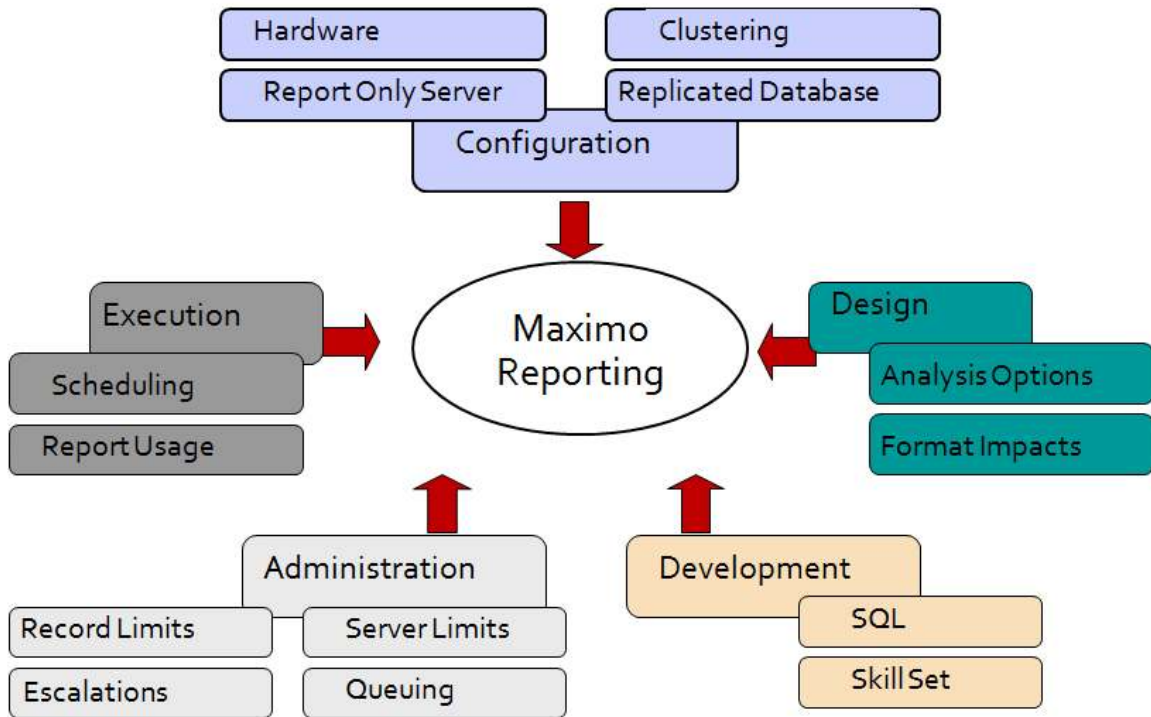
Administering

Administration of reports is very critical to their performance success. Administrators should review include defining security group limits for Ad Hoc reporting, scheduled reports and report server limits. Additionally enabling record limits on critical reports can prevent users from inadvertently executing reports against large data record sets. Additionally, administrators can use automated notifications and receive notices on long running reports before negative performance impacts are realized.

Running

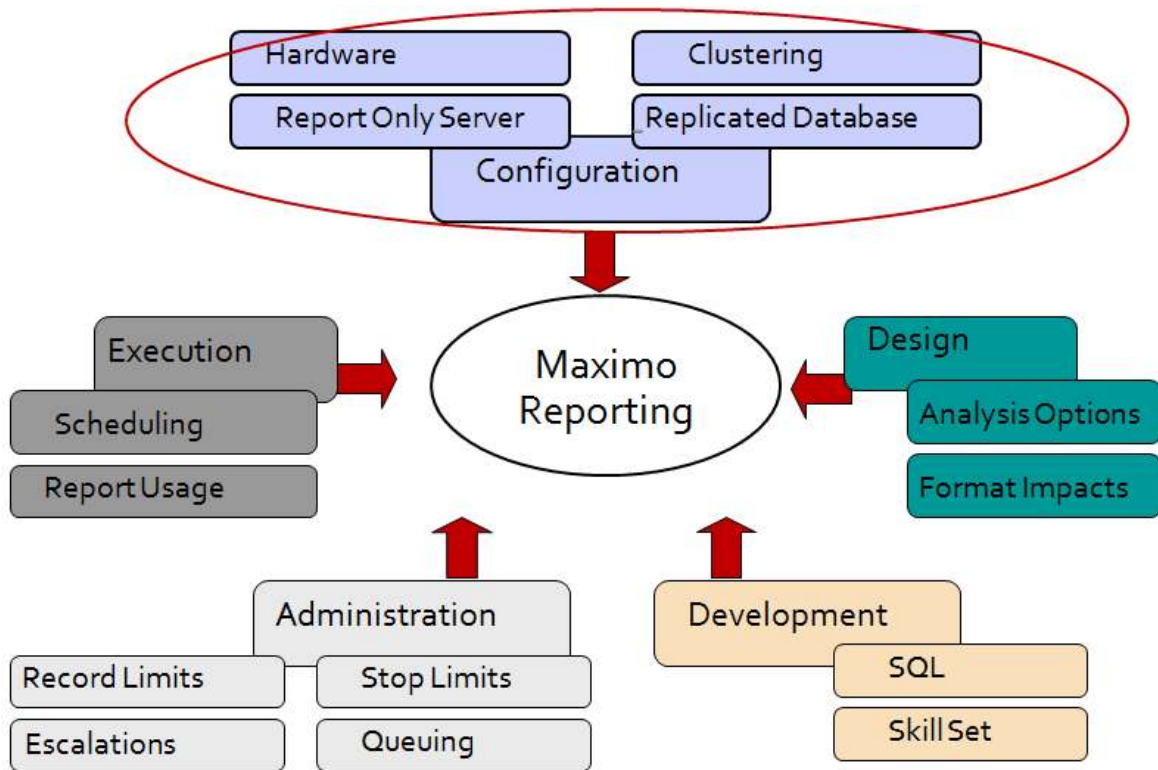
End Users can optimize report performance by scheduling complex, batch reports to execute at off-peak hours. Additional features are available to users to manage their own reports to insure optimal performance.

For each of the five components, segment(s) of each will be explained in more detail below.



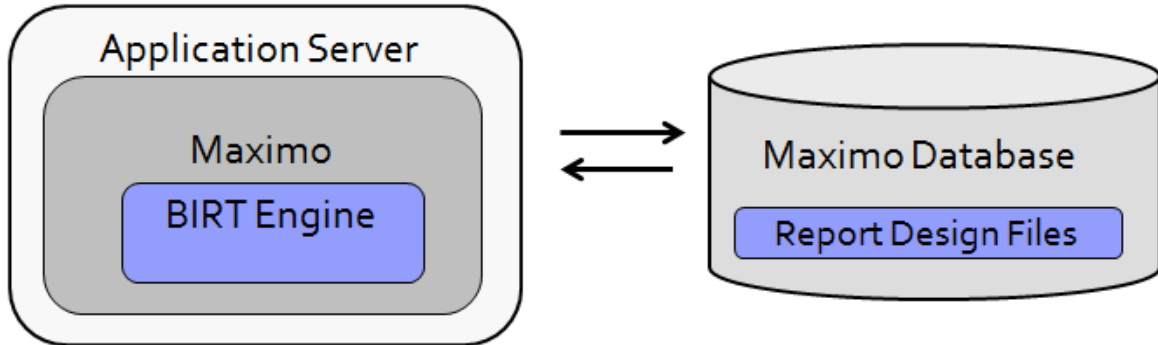
1 Configuration

This section reviews report configuration features available to you. Reviewing the configuration of your Maximo environment - including clustering, report only server, replicated databases and hardware - can have a major impact on your report performance.

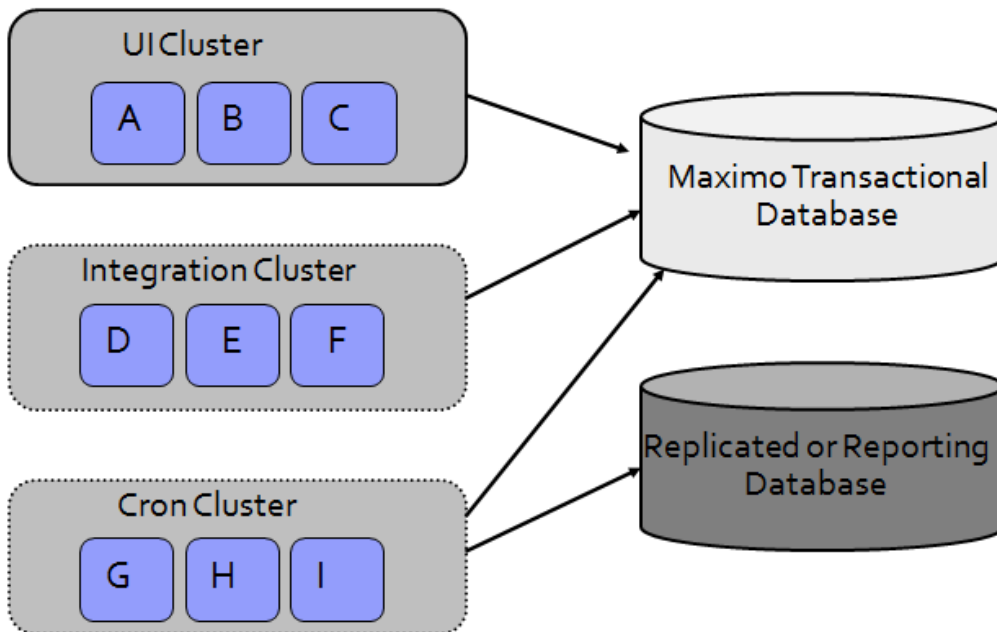


1.1 Configuration: Clustering

In a very simple development or test environment, the Maximo application may be deployed on a single application server. This environment is then configured to a single Maximo database.



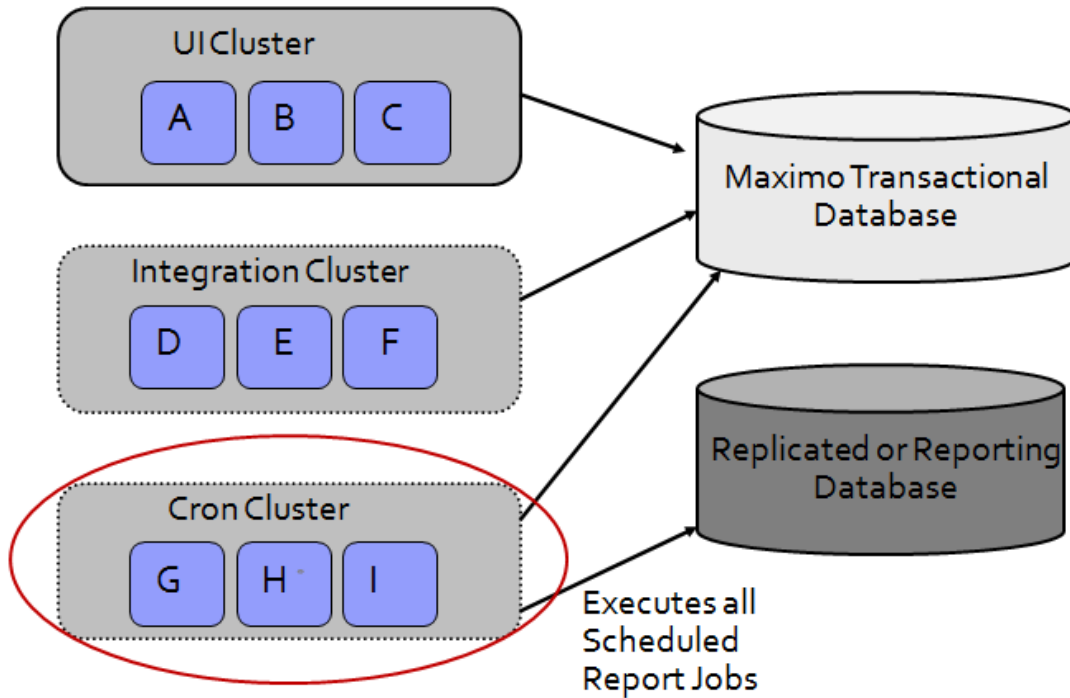
Advanced Server Configurations are required in production environments to enable the system to scale effectively and enable the best performance for all users. At a top level, three types of clusters are recommended which are UI, Integration and Cron Clusters.



UI Cluster: The UI cluster is intended for users to access the applications from a Web Browser. This cluster is typically set up for load balancing so that users accessing this browser only use one URL to access Maximo that may be running on any one of the servers within this cluster.

Integration Cluster: This cluster processes integration messages from message queues, and moves messages into the queues by using HTTP Post, Web Services and EJB Technology.

Cron Cluster: The Cron cluster processes scheduled jobs. These scheduled jobs can be related to integration and reports. This section will focus on the Cron cluster, and enabling it to executing all the scheduled report jobs.

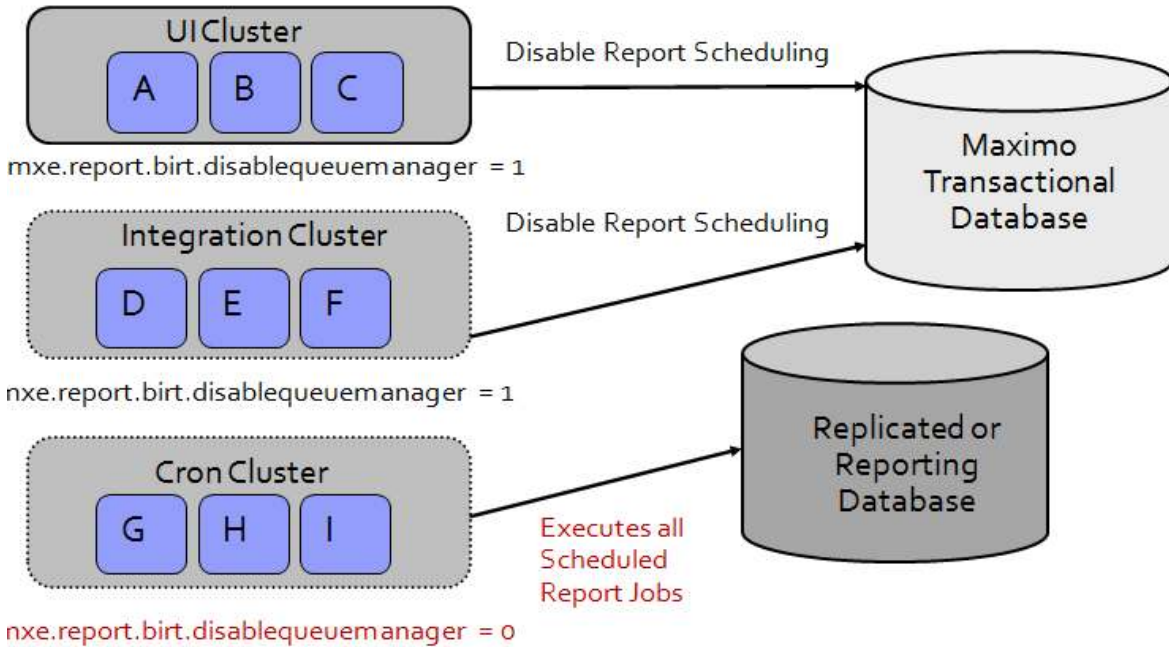


To optimize performance in a clustered environment, it is recommended that all scheduled report jobs be executed from the Cron cluster. This will divert the scheduled report load from the UI cluster to the Cron cluster. This diversion will increase the response time users will experience in the UI cluster.

Users in the UI cluster can still execute immediate reports to receive the immediate report information they need. However, the more complex, analysis type reports that are scheduled will be off-loaded to the Cron cluster of servers.

To enable report scheduling to only occur within the Cron Cluster, a property value needs to be configured. The property value is `mxr.report.birt.disablequeuemanager.*` This value should be set to 1 for any server where report scheduling is not enabled.

If the queue manager is disabled (value set to 1) scheduled reports will not be executed on the server.

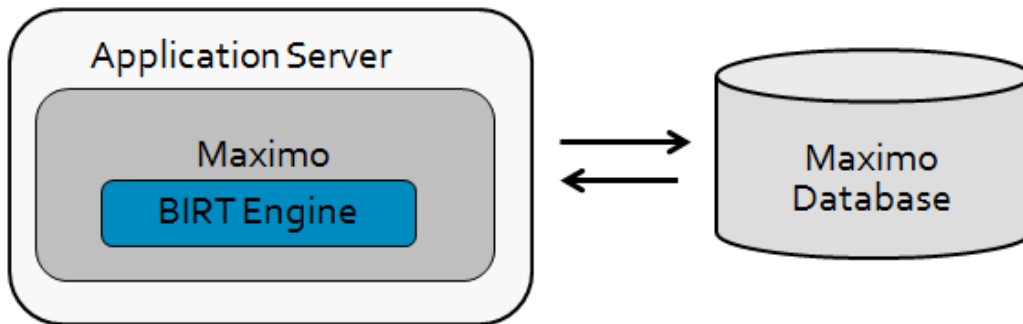


More details on clustering and enabling its setup are contained in the Maximo System Administration Guide, in the System Configuration Chapter.

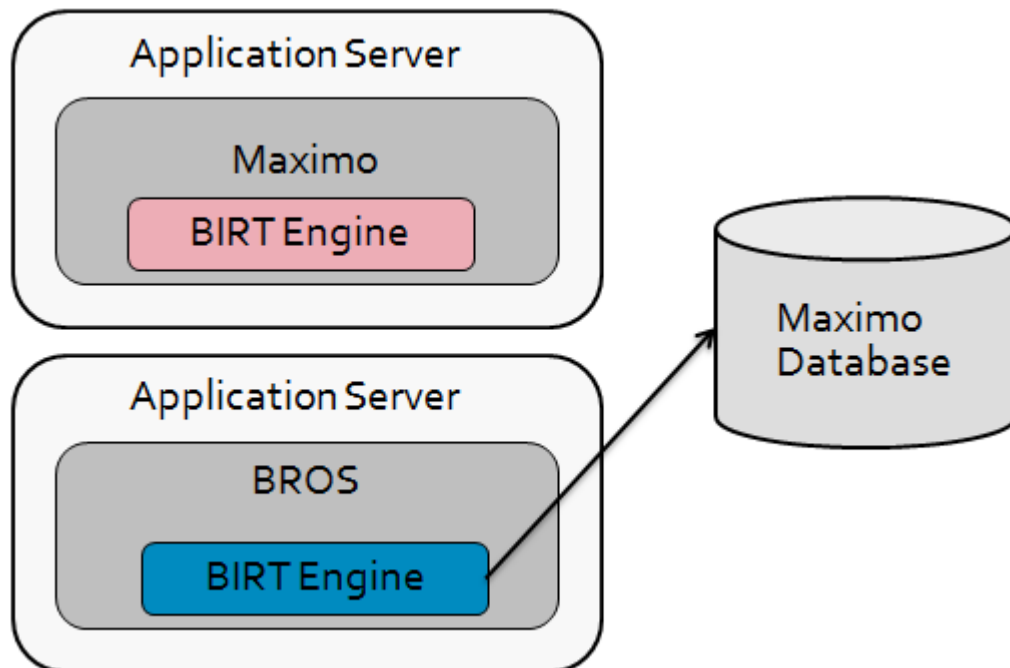
1.2 Configuration: BIRT Report Only Server

You can configure your Maximo environment to have a BIRT Report Only Server (BROS). Enabling BROS will balance report load processing and can improve overall system performance. The BROS will be utilized for report processing – regardless of what clustered server the user may be on.

To highlight this functionality, simple test or development environment is shown below. In this configuration, the Maximo Server is deployed within an application server. This deployment automatically installs BIRT through the ear file process. When a user performs any Maximo action or executes a report, both processes occur within this same server.

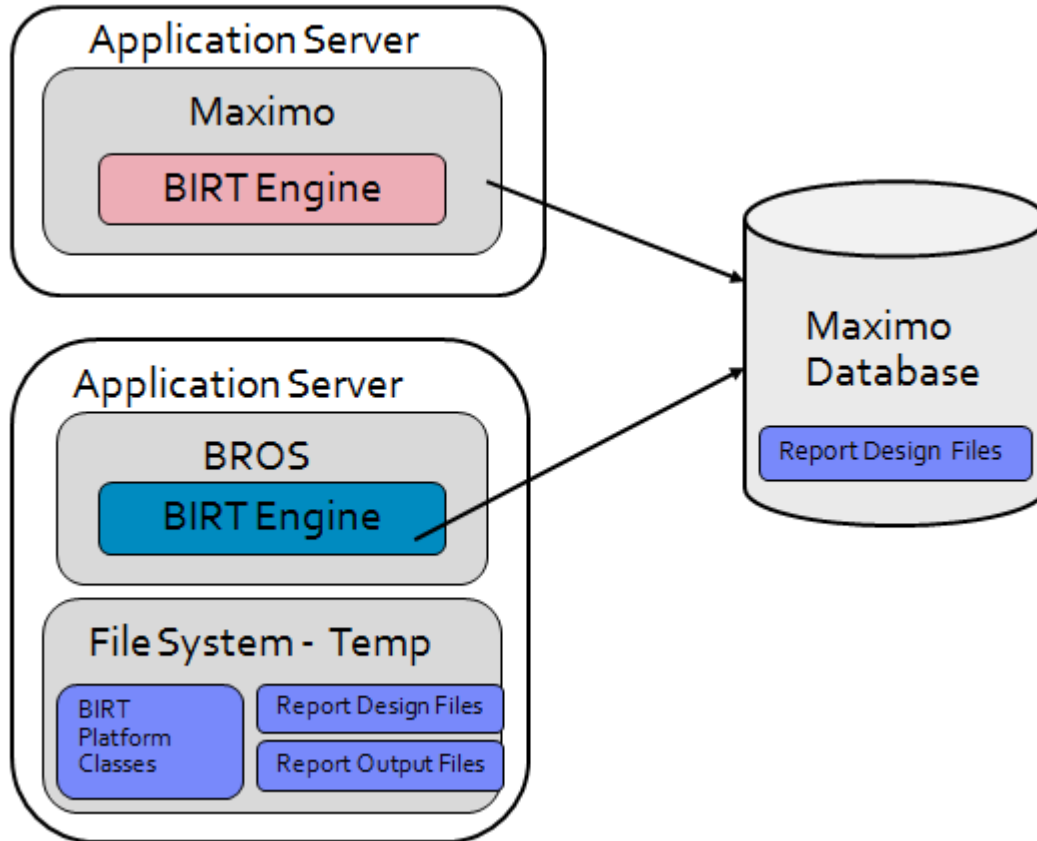


You can extend this environment to configure an additional server to handle reporting processes. This will remove potential performance impacts by off loading reporting processes to a different server.



With the BIRT Report Engine on a separate server, the dynamic hardware requirements required for the report temporary files will only be required on the BROS. These temporary file requirements include the copies of the report design files, the report output files and the BIRT platform classes.

Two JVM System properties define the location of the temporary files on the application server. These files are `mxr.report.birt.tempfolder` and `io.tmpdir`. Details on these JVM properties, including how they can be set, can be found here: <http://ibm.co/1a2zRaQ>



To enable the BROS Configuration, a property file will be used to identify the URL of the report server. Its value is: `mxr.report.birt.viewerurl*`

This property setting is not required.

- If its value is not defined, reports will execute from the Maximo Server.
- If the property is defined, reports* will only execute from the Report Only Server, and/or the Cron Task Server (If Report Scheduling is off-loaded to a Cron Cluster).

The diagram below shows an example of how this could be configured. In this example, there are two servers:

Maximo: UI server used for execution of Maximo applications

BROS: Used only for report processing of Maximo reports

To enable this configuration, two property settings are required.

The first global property setting points to the location of the BIRT Report Only Server,

`mxe.report.birt.viewerurl`.

This value is set to the location of BROS: `http://185.231.245.4:7001`

The second property setting determines if report scheduling should be executed from the server. This is property setting `mxe.report.birt.disablequeuemanager`. In this example

For Maximo, this value is set to 1 (Yes) because no report scheduling should occur on this server.

For BROS, this value is set to 0 (No) because report scheduling should occur on this server.



`mxe.report.birt.viewerurl = http://999.888.777.6:7001`

`mxe.reprot.birt.disablequeuemanager = 1`
(Disables report scheduling from Maximo Server)

`mxe.report.birt.dispablequeuemanager = 0`
(Enables reports to execute from BROS Server)

Notes:

1. The hostname in the Maximo URL cannot be the same as the hostname in the BROS URL. For example, if the user logs into Maximo as <http://abc.com:9080/maximo>, the hostname is "abc.com". The BROS value, `mxe.report.birt.viewerurl`, *cannot* also use abc.com.

- If the same hostnames are used, the Maximo UI Session will become invalid, and the user will be logged out.

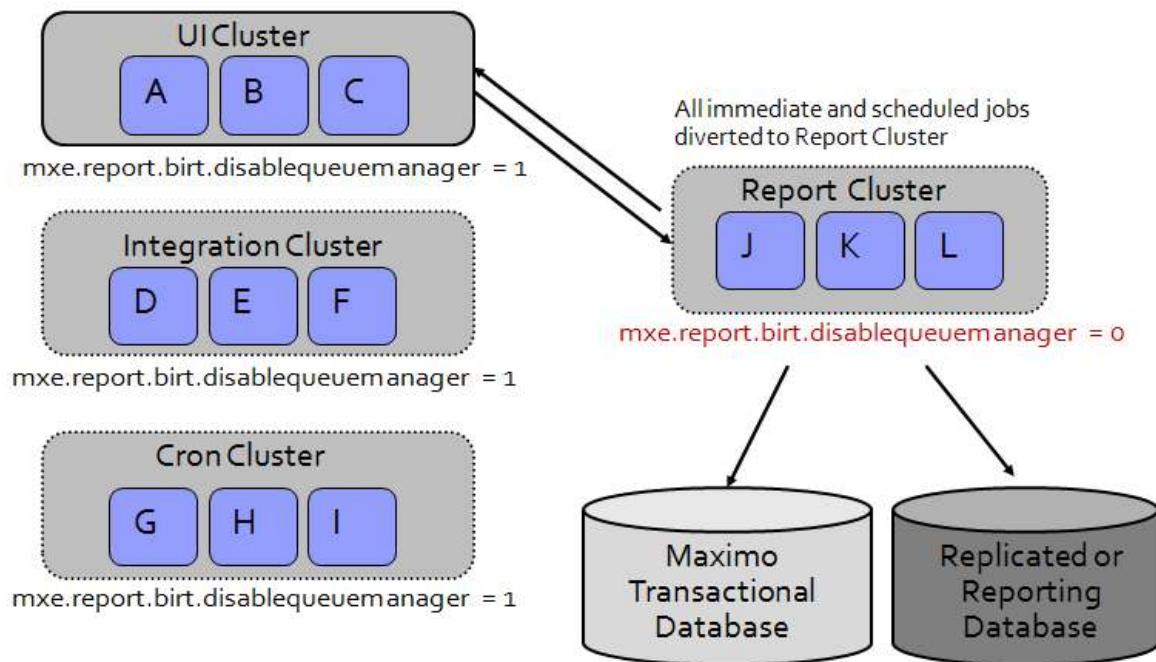
2. When a BROS is configured in a your environment, the BIRT Engine will still be deployed on the Maximo Server. However, it will not be used.

3. The examples used here are for non-SSL Enabled Environments. If your environment is SSL enabled, be sure to use https in your url values.

With the BROS configuration, you can enable a number of different clustered server arrangements. One arrangement that would be recommended is to extend the three cluster configuration (UI, Integration and Cron) to a four cluster configuration, which would include:

1. UI
2. Integration
3. Cron
4. Report

This configuration could improve overall system functionality by removing the report processing load from the UI Cluster. Additionally, any risk of long running or complex reports causing the report server to hang - would be removed from the UI servers. All report jobs – both immediate and scheduled – would be diverted to the Report cluster. This configuration is shown in the diagram below. Each of the four clusters contains three servers noted by an alphabetical character.

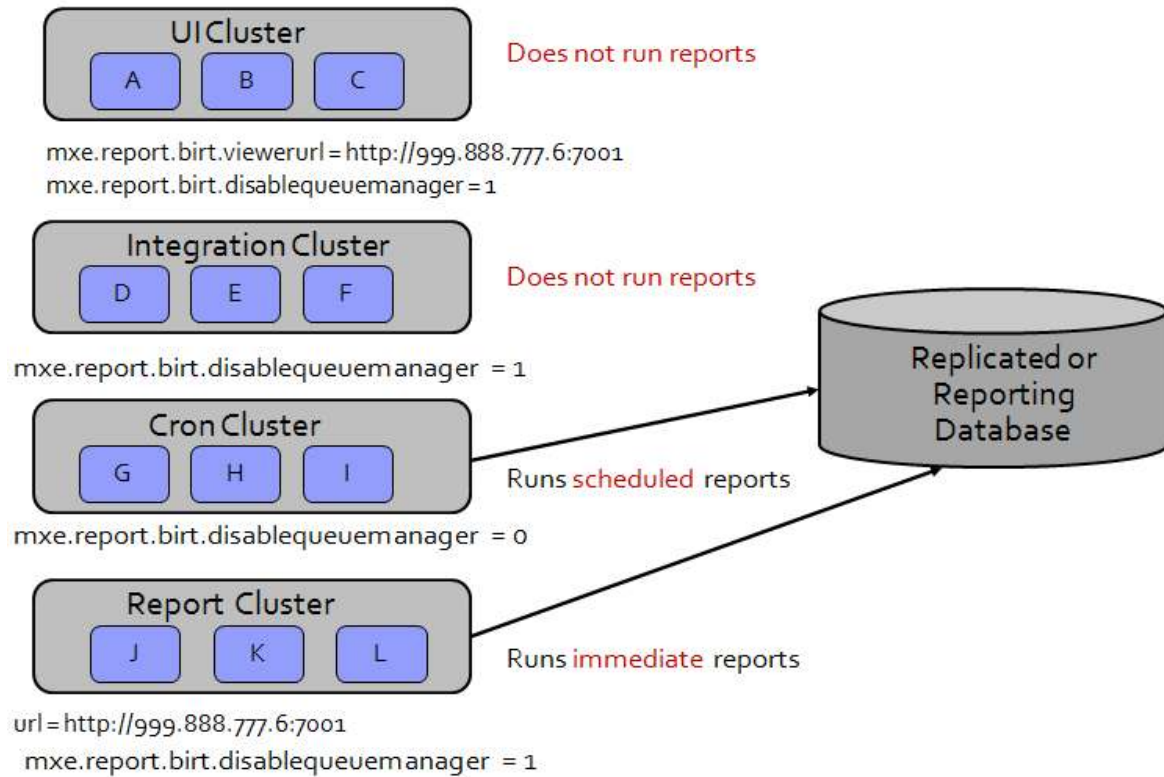


In order to enable report jobs – both scheduled and immediate – to run off the Report Cluster, the Report Queue Manager has to be disabled from the other servers. Disabling the Report Queue Manager prevents scheduled reports from executing on that server. (More information on the Report Queue Manager is contained in the Report Feature Guide referenced at the end of this document.)

To disable the Queue Manager, the property `mxm.report.birt.disablequeue manager` must be set to 1. In this example, the UI, Integration and Cron Servers all have this property set to 1. Additionally, in this configuration, the server where Maximo is deployed (UI Cluster) would have a defined URL for the `mxm.report.birt.viewerurl`. The URL would point to the location of the BROS.

Multiple servers can be used for report servers in a clustered environment. In the diagram above, three Report Servers (J, K and L) are included in the Report Cluster. However, to enable this, only one report server URL is required. This report server URL essentially points to the load balancer, which directs the report job request to any number of report servers defined within the cluster.

In another example of a clustered environment, four clusters are still used. However, in this case, the report scheduling will be done in the Cron Cluster, and the immediate report processing will be done in the Report Cluster. Both the cron and report clusters are configured to use the replicated reporting database.



This will again remove the load from the UI cluster, and place the report processing in both the Cron and Report cluster. This arrangement would be ideal if you process a large number of immediate report jobs as the report cluster would fully focus on processing immediate jobs only.

In order to enable the scheduled report jobs to execute from the Cron Cluster only, the Report Queue Manager has to be disabled from the other servers. Disabling the Report Queue Manager will prevent scheduled reports from executing on that server.

To disable the Report Queue Manager and report scheduling, the property `mxe.report.birt.disablequeue manager` must be set to 1 in the UI, Integration and Report Servers. Only the Cron Servers will have the `mxe.report.birt.disablequeue manager` set to 0 to enable report scheduling on its server. *

Note:

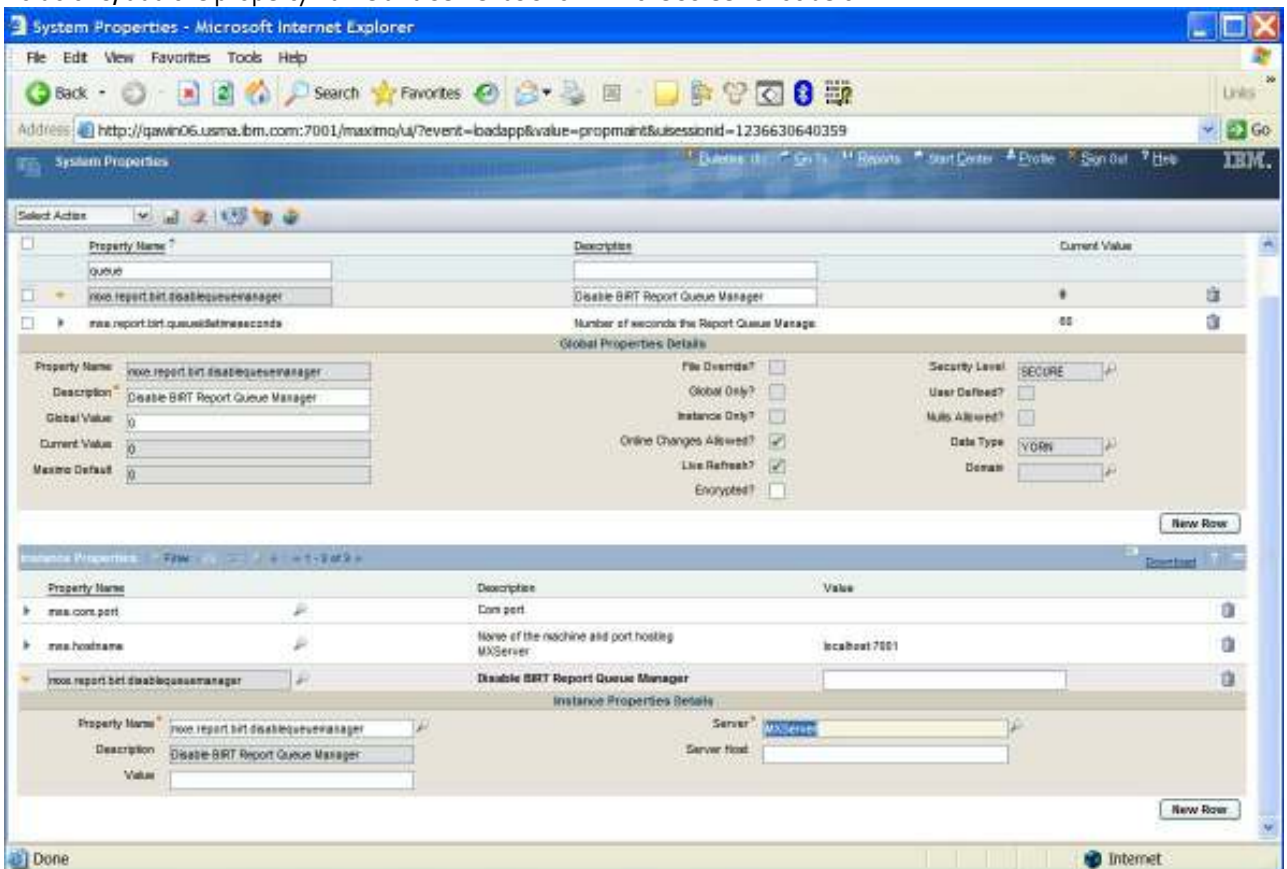
The queue manager property file can be overridden at the server instance level if needed. Information on how to do this is contained in the Maximo System Administration Guide, and is also copied below.

Instance Properties

An instance property is defined and associated with a specific system server. For example, you can configure the system property, `mxe.crontask.donotrun`, to be an instance-specific property. To perform this configuration, access the System Properties application. In the Instance Properties table window, associate the `mxe.crontask.donotrun` system property with a specific server (for example, `MXServer1`) and a value applicable only to that server (for example, a value of `BBCron`). As a result, the Bulletin Board cron task (`BBCron`) does not run on `MXServer1`; however, `BBCron` can run in another system instance, such as `MXServer2`.

When you create both a global value and an instance-specific value for the same property, the instance-specific value takes precedence.

To do this, add the property name and server as shown in the screen shot below.



BROS Authentication

When a BROS is configured, authentication information needs to be passed from the Maximo Server to avoid re-login into the BROS that receives the request to process the report.

This authentication is not required when the report server is on the same server as Maximo as the two share a session.

To enable authentication for the separate Report Server, tokens will be used. This will enable authentication whether the configuration uses the application server authentication or Maximo authentication. The top level steps describing this authentication are:

1. When a report is executed, a random token is generated.
2. This token is stored temporarily in the database table, MAXTOKEN, and is passed to the BROS.
3. The report process servlet(s) receives the token and verifies it with the database.
If the verification is successful, an authenticated session is populated in the HTTP Session, and the token is removed from the database.

Subsequent report requests will then check for the authenticated session information and process the request.

Use cases may result where a token is generated and stored in the database, but the report process servlet does not get a chance to process the report request. In this case, token values can accumulate in the database. To prevent this from happening, token records will be cleaned up via a configured property setting, called `mxe.sessiontoken.timeoutseconds`. By default, this value is set to 180 seconds, or 3 minutes.

1.2.1 Configuration Recommendations and Notes:

1. The BROS server does not support application server security. As noted above, BROS utilizes token based functionality for authentication.

However, LDAP is supported with BROS and Maximo if the domain name of BROS and Maximo is the same. For example if maximo's url is abc.xyz.com and the BROS url is def.xyz.com – then, this configuration will work as they have the same domain name.

2. Enabling the BROS Configuration does not change the Maximo installation or ear file deployment process. The same configurations are deployed on all servers. For example, the entire Maximo applications are still deployed on the BROS Server – however, only the reporting functionality is be enabled.

3. When the report server is deployed inside a Maximo server, it is recommended that no more 5 reports run concurrently at any given time. This value is defined via the property file, mxr.report.birt.maxconcurrentrun.

The value of 5 is based on Maximo server scaling of 50 concurrent users per server. Estimates indicate that that 10% - or 5 users – will be running reports concurrently. Therefore, the value of 5 is used.

However, if BROS is configured, the prior maximum number of 5 concurrently running reports may no longer apply. In this case, it is recommended to increase the property value, mxr.report.birt.maxconcurrentrun, to '2* Number of CPUs' of the Server.

4. Also, multiple clients have asked that since Maximo is scaled for 50 concurrent users per server, does this mean that the BROS can also handle 50 concurrent report users?

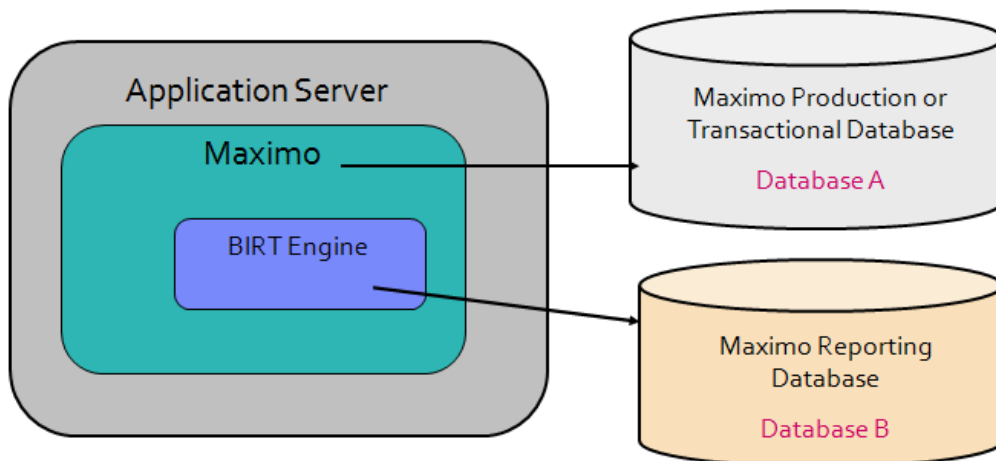
For example, a client has 500 Maximo Users. He uses 10 Maximo Servers to handle the 500 users. Assuming the standard 10% of those users at any time are concurrent report users, can 1 Report Server handle the 10% or 50 Concurrent Users running reports from these 10 different Maximo servers?

The answer to this is not straightforward as it depends on the types of reports that are being run. Reports have a huge range in complexity, from a very simple list report to a very complex Maintenance Cost Rollup report that encompasses numerous subreports and hierarchy levels. Therefore, this answer depends on the average complexity of reports that are being executed.

1.3 Configuration: Replicated, Reporting Databases

You may want your reports to execute from a separate, reporting database. This separate, external database may be a snap-shot of their production, transactional database, and is sometimes referred to as an external, replicated or reporting database.

Using reporting databases can reduce the load from your live, transactional database and lead to performance improvements. Additionally, if these reports are scheduled, additional processing time can be off-loaded from the Maximo Server. These performance improvements can be especially noticeable for complex reports.



This type of configuration can be enabled two different ways:

1. All reports can be configured to execute against a reporting database,
2. Or a portion of the reports can execute against the reporting database.

To enable these two different types of configuration, the Reporting Database (Datasource) needs to be registered in the Report Administration application. Then, if only a subset of reports will execute from the Reporting Database, additional steps will be required to modify their individual report design files within the BIRT Report Designer.

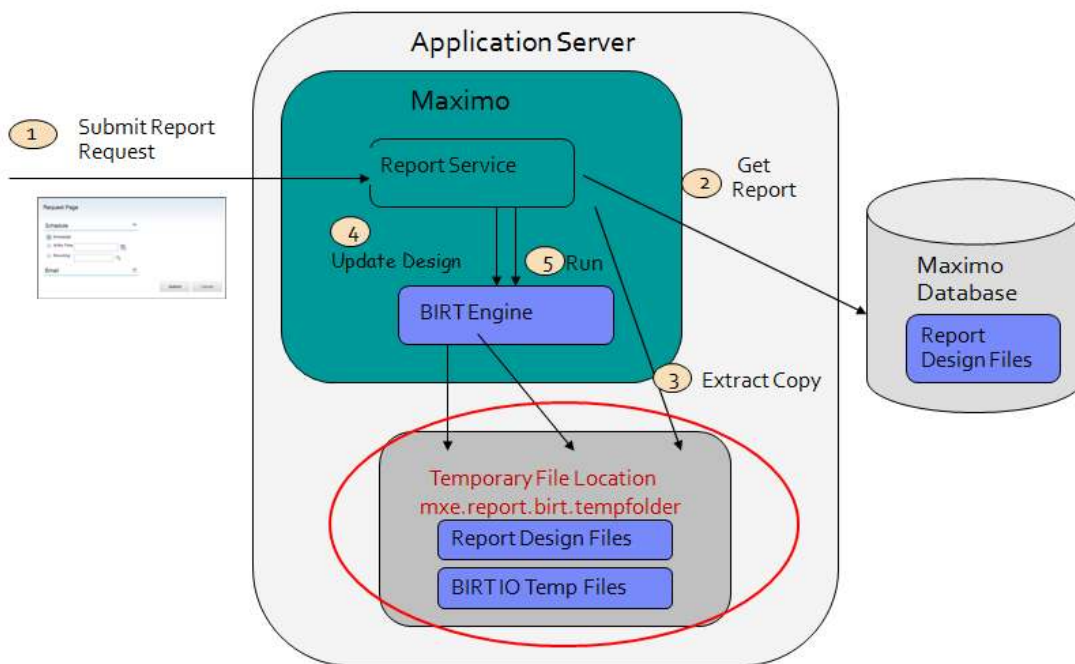
The steps on how to configure each scenario are contained in the 'Enabling Secondary Database Configuration for Reporting' Document which is accessible [here](http://www-01.ibm.com/support/docview.wss?uid=swg21304936), or via its url below.
<http://www-01.ibm.com/support/docview.wss?uid=swg21304936>

1.4 Configuration: Hardware

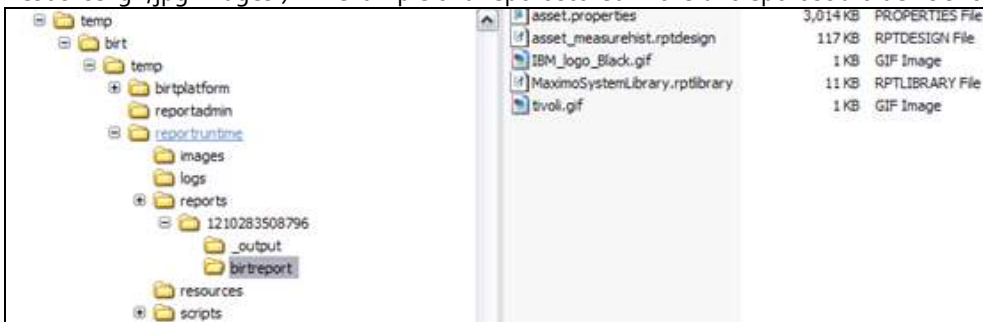
As noted in the Maximo Reporting Feature Guide, temporary disk space is required on each Maximo server executing BIRT reports. This disk space is needed to hold temporary copies of the BIRT design files and graphic images, along with the BIRT Platform Classes.

The BIRT Platform Classes are required in the temporary location for initializing the BIRT Report Engine. These files are approximately 24 MB, and will always be required.

The report design files and the report output files are also stored in the file system's temporary directory as highlighted within the red circle below. These temporary files are required when users are actively running reports.



The report design files are the temporary copies of the user's requested report from the database. This will include not only the .rptdesign file, but also any libraries or resource files it uses (ex. Property files and Resource .gif/jpg images.) An example of a report stored in the birtreport subfolder is shown here:



The report output files are the actual reports that are generated based on the user's individual request. These files have the extension of .rptdocument, and are stored in `<c>temp\birt\temp\reportruntime\reports_output`



The amount of file space required on your Maximo environment for the temporary report design and output files depends on your individual business environment. The file space required depends greatly on the:

1. File size of the report design and its output
2. Whether or not the report contains graphic images or not
3. How many reports are being concurrently run by users
4. How long the Maximo session is kept open

These temporary files are retained until a cleanup mechanism deletes them. This mechanism is invoked according to certain business rules, including:

- A. If the user runs the same report multiple times, the previously run report information from the temporary folder will be removed.
- B. If the user signs out of Maximo, then all the temporary report information related to all the reports that the user has run from the browser will be removed.
- C. If the user session times out (which is by default, 30 minutes), then all the temporary report information related to all the reports that the user has run from the browser will be removed.
- D. If the server is restarted, then all the temporary report information related to all the reports that any user has run from the browser will be removed.

This example highlights the variability of the space required. An environment has 500 Maximo Users. Of those 500 Users, 10%, or 50 Users are concurrently running reports.

Each of the 50 users executes a report that has an output of 1MB.

*For these 50 users, 50MB of disk space is needed. (50 Users * 1MB)*

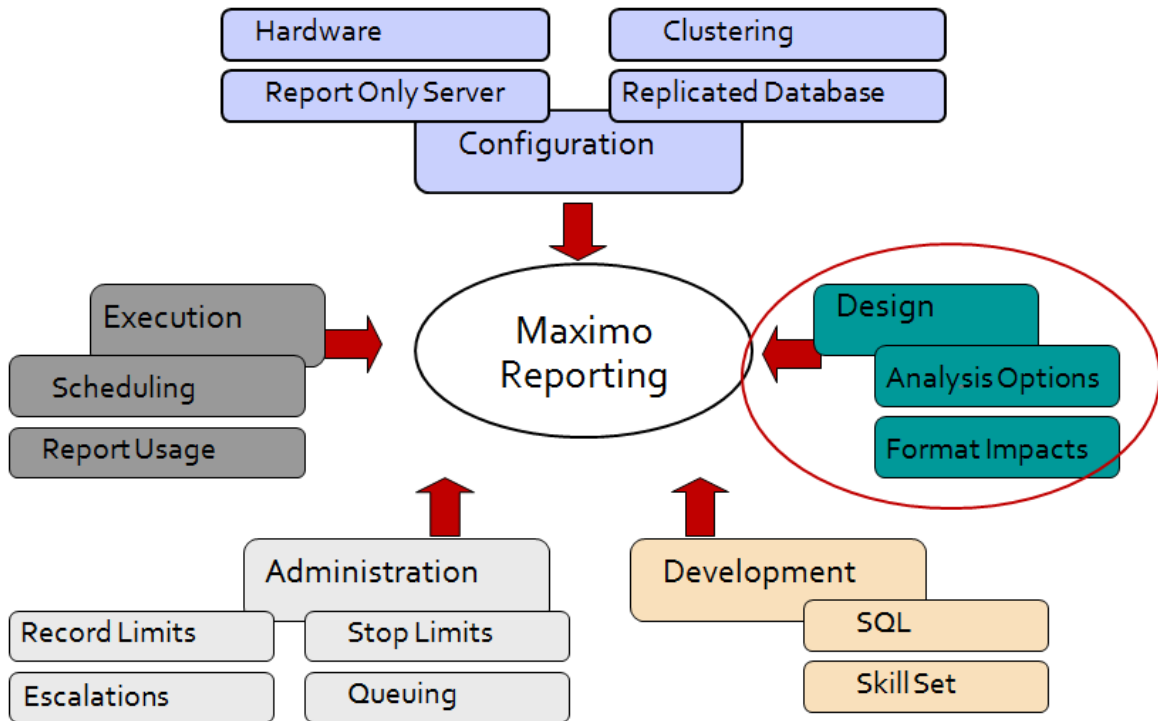
Now, imagine each of the 50 users each run 5 reports.

*For these 50 users, 250MB of disk space is needed. (50 Users * 5 Reports * 1MB)*

As this example shows, the amount of temporary file space you required will vary depending on the number of active report users you have along with the report file sizes.

2 Designing

The report design process is frequently overlooked when analyzing report performance. However, a report's design and format - has a major impact on how quickly reports are rendered to the user.



2.1 Design: Analysis Options

Within the various applications, there are a variety of mechanisms available to analyze the powerful data that is generated. These mechanisms present the data in a variety of formats, so you can quickly get the information needed to make the best business decisions possible.

Depending on the individual business case and user, you may want to evaluate using different Data Analysis Options than reporting in certain cases. For example, one of your users may need to know the Number of Work Orders Overdue each day when he logs in. This information is critical for him to meet his job objectives, and he wants to quickly see this data. In this case, a KPI visually displaying the Number of Work Orders Overdue on the user's Start Center may be a better fit to convey this information.

When a report is created, it must first be designed, developed, tested and maintained. Depending on the complexity of the report, this could take a few hours, a few days or a few weeks. Then, once those reports are available, they will require system resources to execute. Again, depending on the complexity of the report, those system resources could be minimal or significant.

The latest data analysis features available in Maximo are listed below. These features enable different forms of data analysis, which can quickly and dynamically display data in any applications. Additionally, they can be considerably less time consuming to enable than creating individual custom reports, and therefore should be carefully considered.

The different functionality formats include:

QBE – Query By Example. Using your application's filter and/or query, you can immediately download your results for additional analysis in Microsoft Excel.

RS – Result Sets. Using an application's query, you can enable a set of fields or graph for display on the Start Center. Starting in Maximo 76, result sets can utilize object structures, enabling you to select attributes from multiple related objects.

Application Export. Capitalizing on the power of an object structure, you can export multiple database objects and attribute fields using your application's filter and/or query.

KPI - Key Performance Indicators. Visual indicators displaying status against predefined targets. Starting in Maximo 76, KPIs have been extended with new KPI Template and KPI Viewer applications, along with the ability to schedule individual reports, and define which users have access to them.

QBR – Query Based Reporting. Maximo's Ad hoc reporting tool empowers your users create their own reports from within the various applications. Users can select attributes from multiple objects, apply application queries, and define sorting and filters. Additionally, in Maximo 76, power users can apply summaries and define calculations for their ad hoc reports.

OR – Operational Report. Often referred to as transactional reports, these are the traditional day to day detail reports your users require to complete their business tasks.

SR – Strategic Report. Thru the use of complex graphs, in depth calculations or scenarios, these reports enable analysis of data in varying perspectives. Additionally, the provide user interaction in dynamic filtering and display of the data.

For additional details on these data analysis options, reference the Maximo Report Upgrade Planning Guide available within the report documentation referenced at the end of this guide.

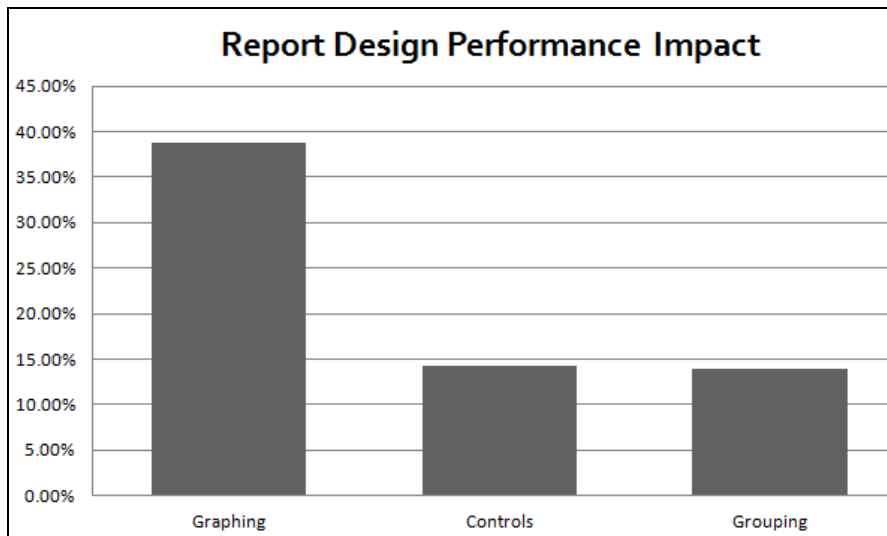
2.2 Design: Format Impacts

The specific format designed into a report has a significant impact to its performance. Reports that contain multiple graphs, groupings and controls will require a longer time to execute than simple listing reports.

To highlight the impact of formatting, a variety of tests were performed using the Maximo delivered reports in the Maximo 7.1x release. This testing was done to quantify the performance impact of graphs, groupings and controls.

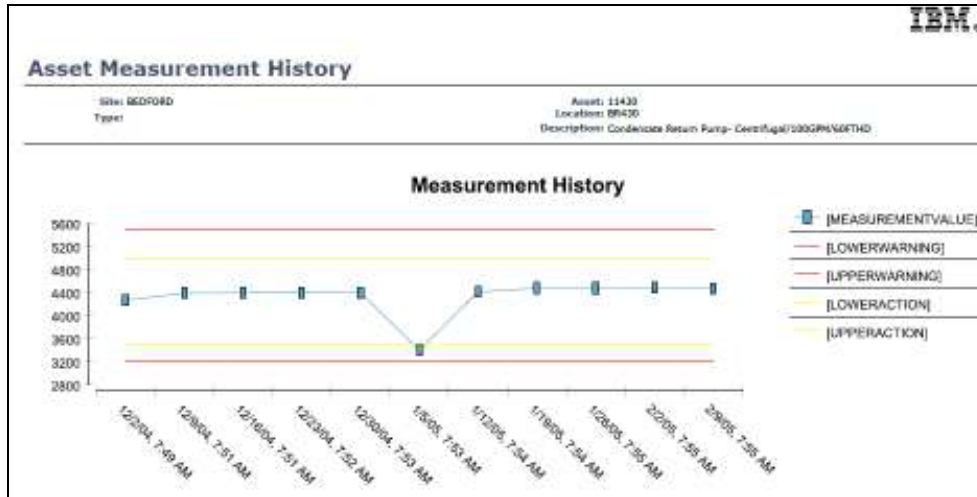
The chart below shows the largest performance impact seen by reports that included graphics, followed by those with a large number of controls and multiple groupings. While hundreds of reports were executed to determine these results, your individual results may vary depending on the specific designs of the reports tested.

Each of these tests was performed in the same Maximo environment. A variety of users executed these reports from different applications and from different access points.



Graphs

Graphs are extremely useful in quickly conveying data results to users. Their visual representations often immediately highlight results, so users do not have to manually analyze the data themselves.

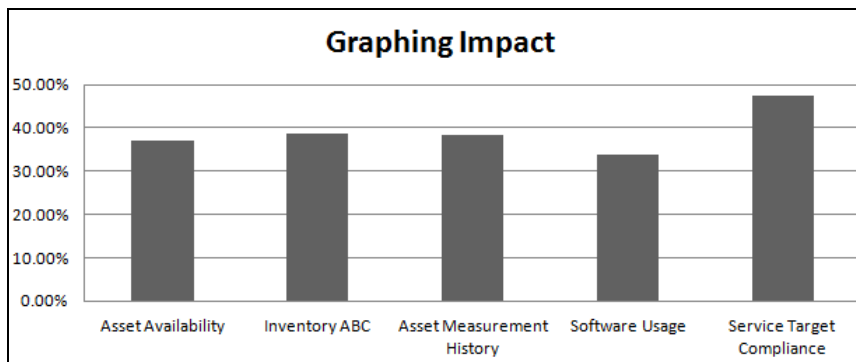


To quantify the impact of graphic rendering on report performance, five different OOB Reports were analyzed, including Asset Availability, Inventory ABC, Asset Measurement History, Software Usage and Service Target Compliance. These reports included a variety of graph types, including line, bar and pie charts. Each report included 1 graph, except for the Service Target Compliance Report which contained 3 Graphs.

To perform the test, each report with the graphic was executed 10 times. Then, the graph from each report was removed, and the reports were then executed 10 more times. These results were captured in the REPORTUSAGLOG Table.

The results below show that on average, reports with no graphs ran 38.8% faster than those same reports that contained graphs.

These results highlight the significant impact of graphs on performance. This data can be used to validate that you may only want to use graphs in reports that are analysis type reports. Analysis reports are typically scheduled or executed against non-Transactional Databases. Including graphics in transactional reports that need to be executed immediately and consistently throughout the day will have a negative performance impact.



Groupings

Groupings are used in reports to separate data. For example, if you are analyzing an asset list report, the assets may be grouped by Site and/or Location. This enables the user to break down the information. So – instead of evaluating 100 Assets, the user can break it up into 40 Assets in Site Bedford, 25 in Site Texas and 35 in Site Buffalo.

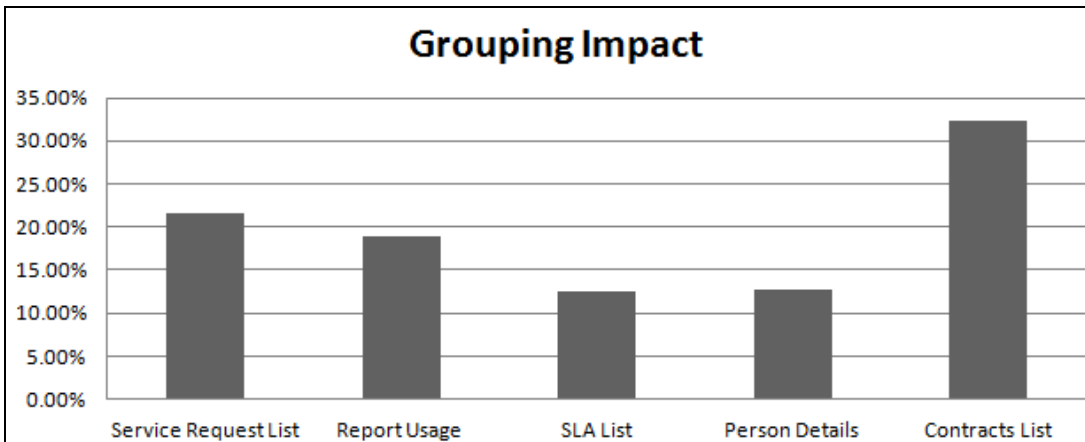
Item Availability									
Organization: ENGLDA									
Item: 11453		Description: Seal, Mechanical, Self Aligning- 1 In ID							
Site: BEDFORD		Storeroom: CENTRAL			Qty Avail: 12		Qty Res'vd: 4		
Bin	Condition Code	Lot	Exp Date	LIFO/FIFO Item?	Current Balance	Issue Unit	Standard Cost	Average Cost	
A-4-9		A-421	Jan 1, 2015, 12:00 AM	N	56	EACH	129.80	130.45	
G-5-2		A-421	Jan 1, 2015, 12:00 AM	N	1	EACH	129.80	130.45	
Site: BEDFORD		Storeroom: RACHSHOP			Qty Avail: 1		Qty Res'vd: 0		
Bin	Condition Code	Lot	Exp Date	LIFO/FIFO Item?	Current Balance	Issue Unit	Standard Cost	Average Cost	
H-9-2				N	1	EACH	129.80	130.45	

To quantify the performance impact of grouping in reports, five different reports were analyzed. These reports included Service Request List, Report Usage, SLA List, Person Details and Purchase Contract List. Two of these reports included two groupings, Service Request List and Report Usage, while the remaining three had one grouping.

Each report was first executed 10 times. Then, the groupings from each report were removed. The reports with no groupings were then executed 10 more times.

The results below show that on average, reports with no groupings ran 14% faster than those same reports that contained groupings.

The results below highlight the impact of groupings on reports. Although this is not as large an impact as graphs, it is an important factor to consider when specifying one or multiple groupings in a report's design.



Report Controls

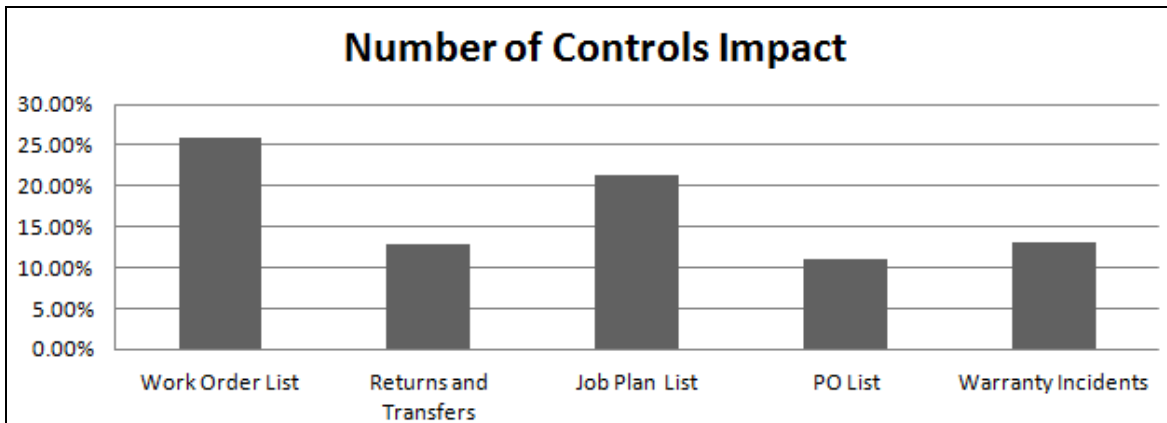
Every report contains multiple controls. These controls could be a database field, calculated value, graphic image or stationary line used to separate data results. The more controls designed in a report, the denser the report output becomes. And the denser a report, the more time it takes to execute. The screen shot below shows a dense report with 18 field controls in a single data row.

Receipt and Transfers Transactions																		
Stations: CENTRAL																		
Site: BEDFORD																		
Start Date: 1/1/00																		
End Date: 1/1/04																		
Time: 11:453			Description: Seal, Mechanical, Self Aligning- 1 In ID									Transaction Type: RECEIPT						
Date	Qty	From Location	To Location	From Lot	To Lot	From Bin	To Bin	Condition Code	From Condition Code	Order Unit	Unit Cost	Line Cost	Actual Cost	Loaded Cost	PO Line	PO Slip	Pack/Slip	
12/27/00	2:29:31 PM	10	CENTRAL		CH-4		N-3-B			EACH	129.00	1390.00	135.00	1417.50	A5238		2	

To quantify the impact of the number of controls used in reports, five different reports were analyzed. These reports included Work Order Tracking List, Returns and Transfers Transactions, Job Plan List, Purchase Order List and Warranty Incidents. Each report was executed 10 times.

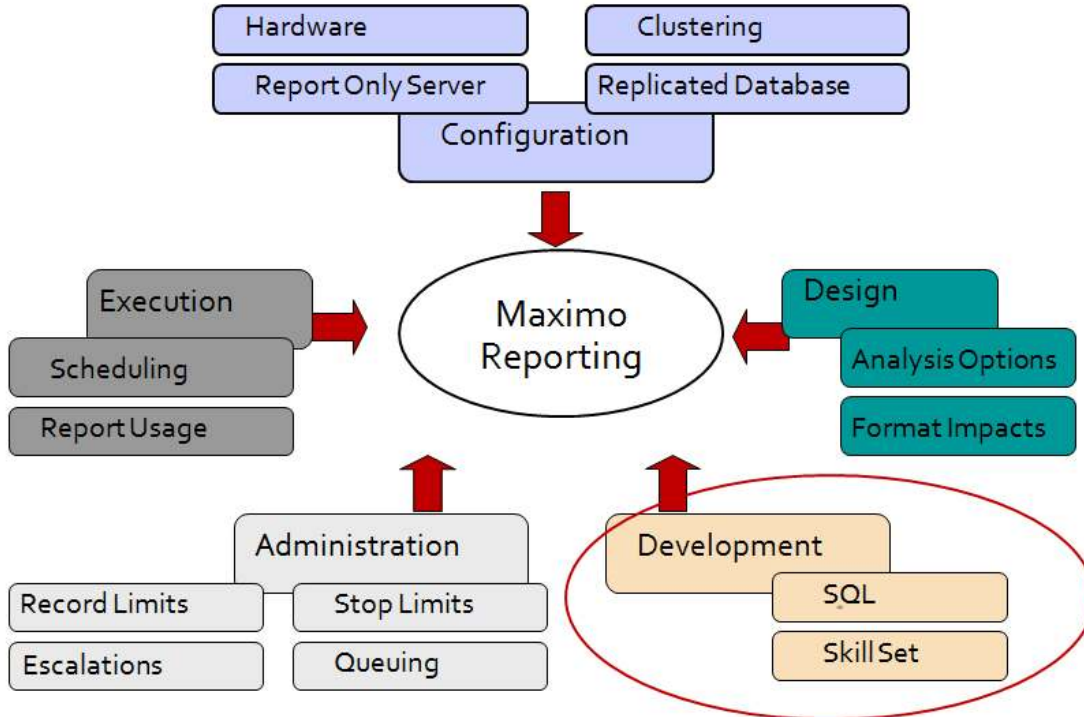
Then, the number of database field controls used in each report was reduced by 50%. For example in the Work Order Tracking List Report, the number of fields shown in the details section was reduced from 12 to 6. (Note: This did not reduce the number of overall controls used in the report by 50% - as many controls remain constant including the Report Titles, graphic images, page counts and date formats.) Depending on the individual report, the number of field controls was reduced from 9 to 4 fields. Once the database field controls were reduced by 50%, the 5 reports were then executed 10 times.

The results below show that on average, reports with reduced field controls ran 14% faster than those same reports that contained additional field controls. These results indicate that the number of fields designed in a report is an important factor of performance.



3 Developing

How your reports are developed - specifically the report's sql statement - impacts a report's performance significantly. Because of this, it is extremely critical that your report developers have very strong skill sets in developing reports, and understanding the Maximo database and object relationships.



Each Maximo report contains SQL (Structured Query Language) which is used to query the database. The SQL details which database tables and fields need to be accessed. When more than one database table is accessed, joins are used to define the relationships between the tables.

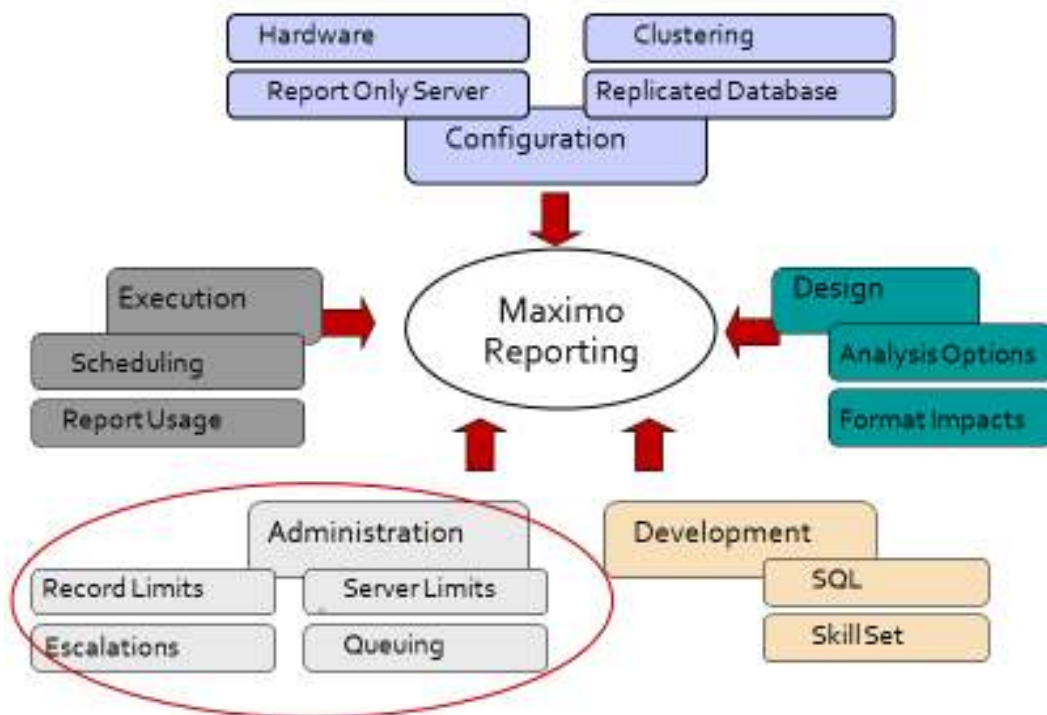
The manner in which the data is retrieved from the database has a very large impact on performance. If a report retrieves data from multiple tables and the joins are incorrectly defined, the performance will be adversely impacted.

The BIRT Designer does not include any tools to either validate the report's sql statement or optimize it. Therefore, it is critical that the developer make sure that each of the report's sql is properly and optimally defined. Depending on the type of database used, the developer can use a variety of tools to do this. Additionally, the developer can verify the joins used in the report by evaluating the various object relationships defined in MAXRELATIONSHIPS, or by reviewing the latest ERD Diagrams noted at the end of this guide.

4 Administration

Multiple actions are available within the Administration process to prevent negative performance impacts and monitor report performance. This can include the configuration of record, report server, scheduling and ad hoc preview limits to prevent users from executing against very large record sets. Additionally, report escalations can be configured to automatically notify the administrator when long running reports are being processed. The administrator can then cancel these reports before negative performance impacts to the server occur.

These administrative features are detailed in this section.



4.1 Administration: Record Limits

Users may inadvertently execute reports against large or very large record sets within an application. This can cause negative performance impacts on both the application and database server.

An example of this is in the Work Order Tracking application. In this case, a user only wants to print out the 10 new work orders which were input during the night shift. However, the user forgot to filter their query, and instead of submitting a Work Order Details Report Request for the only the 10 new work orders, they input a Work Order Details Report Request for all work orders, which totaled over 2500 records. And because the Work Order Details report is a very complex report, this request will tie up the application and database server for some time.

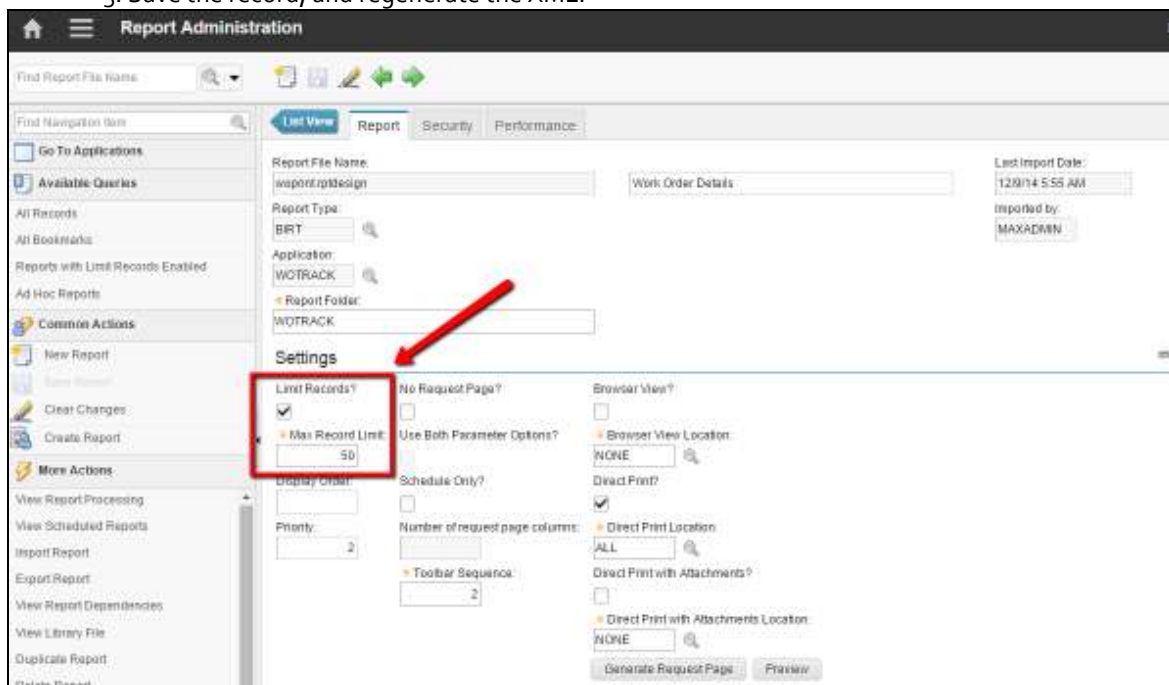
To prevent incidents like this, and from users running large queries, record limits can be enabled on reports. The Record Limit functionality enables you to define exactly the maximum number of records a report can execute against. This functionality is only enabled for reports that use the current/selected record set.

To enable this functionality, a count of the user's current record selection is determined from the application the user is in. This value is then checked against any record limits set for the selected report.

- If the current record set is greater than the record limit, a message is displayed to the user to reduce his query.
- If the current record set is less than the record limit, the report is executed.

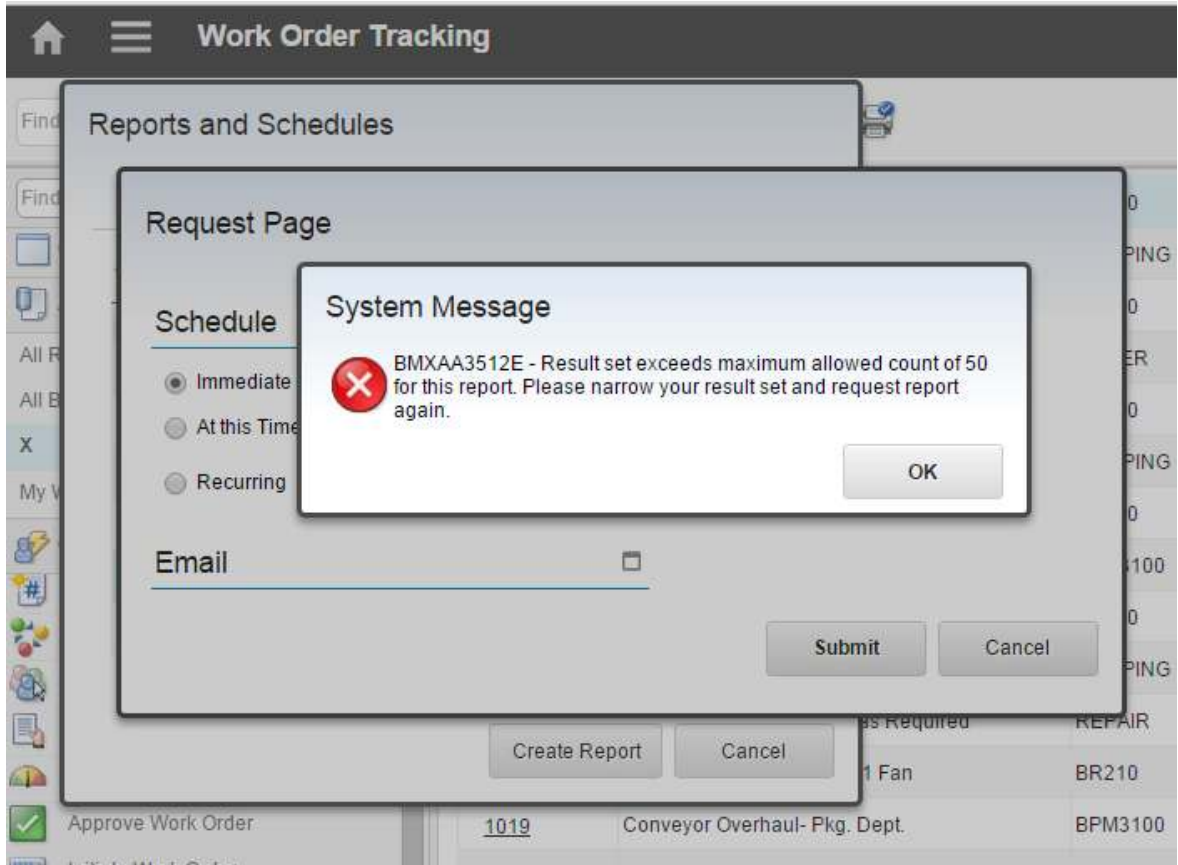
To set up Record Limits, access the Report Admin App.

1. Enable the Limit Record Flag
2. Input a value for the Max Record Limit. The value must be greater than 1, but there is no limit on the maximum number that can be input.
3. Save the record, and regenerate the XML.



If the user tries to execute a report from the work order tracking application against a record set of 100 records - and the report has a limit of 50 - the error message below will display.

'BMXAA3512E - Result set exceeds maximum allowed count of 50 for this report. Please narrow your result set and request report again.'



Note:

1. Only reports using Current/Selected parameters can take advantage of this feature.
2. Beginning in Maximo 76, record limits can also be applied to Ad Hoc reports.

4.2 Security Group Limits: Server Limits

Record limits prevent a user from executing reports against the current record value in an application. However, many reports have multiple subreports, which link to records in child, grandchild, and lower levels of a hierarchy. For example, a work order report can easily have 10 or more subreports to related child objects including Planned Materials, Labor, Services and Tools, Work Logs, Related Records, Actual Materials, Labor, Services and Tools and many, many more related objects.

In these cases, report server records can be configured at the Security Group level. For all or a subset of your security groups, the report server limit value is set to a very high figure. When a report executes, a count of records is made throughout its hierarchy - and if the server record limit is exceeded - the report will stop executing.

Again, this value should be set to a very large figure because of the amount of objects and records that can quickly be included within a single report.

The Server Limit value is defined in the 'Set Security Group Limit' action in Report Administration.

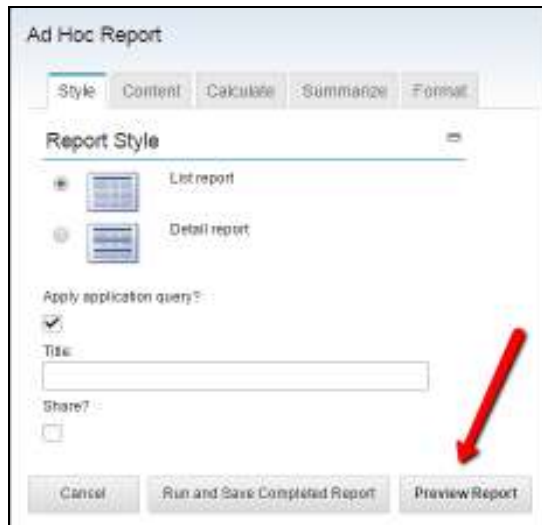
The screenshot shows a dialog box titled "Set Security Group Limits". At the top, there is a search bar with "Security Groups" and a "Filter" button. Below this is a table with two columns: "Group" and "Description". The table contains one entry: "MAINTSUP" with the description "Maintenance Supervisor". Below the table, there are three sections for configuring limits:

- Report Creation:** Ad Hoc Preview Records Limit: 500
- Report Management:** Scheduled Reports Limit: 10
- Report Execution:** Report Server Records Limit: 500,000

At the bottom right of the dialog are "OK" and "Cancel" buttons.

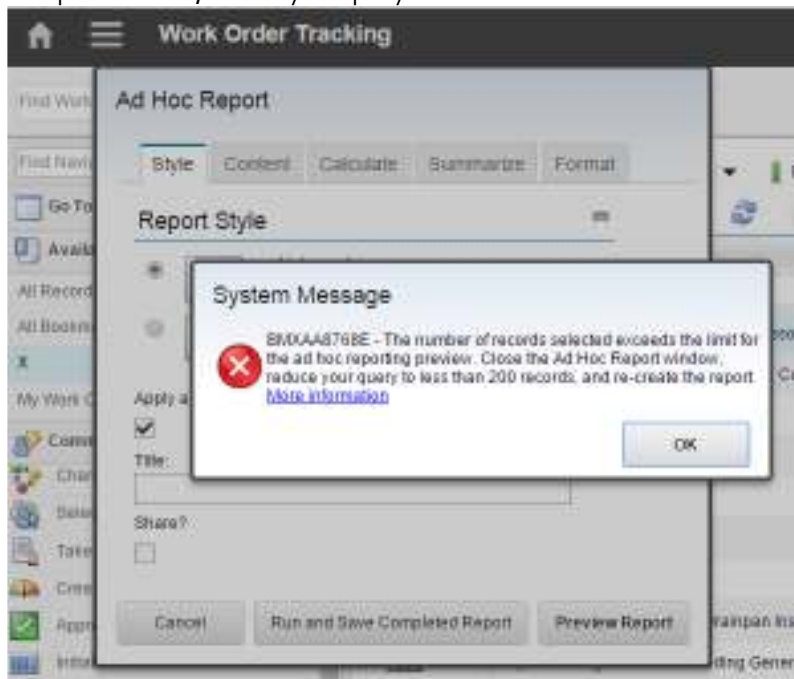
4.3 Security Group Limits: Preview Limits

You may also want to define preview limits for your users creating ad hoc reports. These limits can prevent users from executing ad hoc reports against very large records sets as they build their report content. This is the stage where a user views his report frequently to confirm that the format, fields and calculations are displaying as expected. These reports are often referred to as transient reports.



You may want to set preview limits for a subset or all of your users who have security privileges to create reports. These are defined in the Set Security Group Limits action. If a user tries to preview report with records greater than their ad hoc preview limit, a message will display that they must reduce their query.

'BMXAA8768E - The number of records selected exceeds the limit for the ad hoc reporting preview. Close the Ad hoc report window, reduce your query to less than X records and re-create the report'



Preview limits are applied as the report's record limit when the ad hoc report is saved. The administrator can change/delete these as required.

List View | Report | Security | Performance

Report File Name: ASSET_WILSON_1415672660.rptdesign | Asset Analysis of BR Location | Last Import Date: 11/10/14 8:25 PM

Report Type: BIRT | Imported by: WILSON

Application: ASSET | QBR Created By: WILSON

Report Folder: ASSET | Saved Where Clause: (siteid = 'BEDFORD' and

Settings

Limit Records? <input checked="" type="checkbox"/>	No Request Page? <input type="checkbox"/>	Browser View? <input type="checkbox"/>
* Max Record Limit: 5,000	Use Both Parameter Options? <input type="checkbox"/>	* Browser View Location: NONE
Display Order: <input type="text"/>	Schedule Only? <input type="checkbox"/>	Direct Print? <input type="checkbox"/>
Priority: <input type="text"/>	Number of request page columns: <input type="text"/>	* Direct Print Location: NONE
	Toolbar Sequence: <input type="text"/>	Direct Print with Attachments? <input type="checkbox"/>
		* Direct Print with Attachments Location: NONE

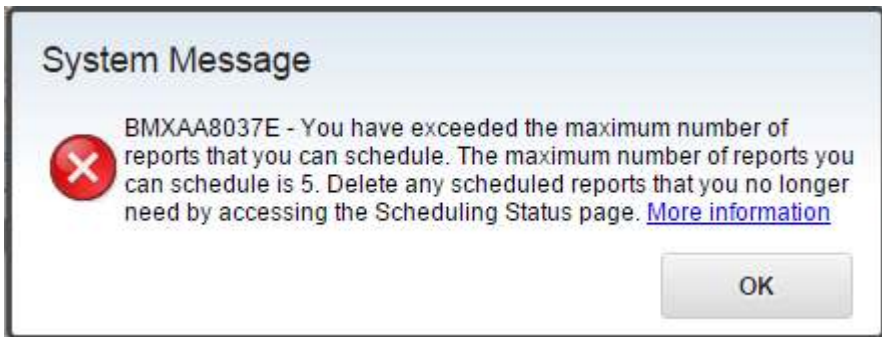
Generate Request Page | Preview

4.4 Security Group Limits: Scheduled Report Limits

You can limit the number of report schedule requests your users submit thru the report schedule limit functionality. By limiting them to a maximum number of scheduled jobs, users will only submit schedule requests for reports they need, and delete requests for obsolete report jobs.

Users may submit recurring schedule requests for reports that they think they will always need – but once the need for the report output goes away – they may forget to delete the scheduled report request. However, the report will still continue to execute. This leads to wasted, unnecessary resources – as the processing continues – while the consumption has stopped.

When scheduled report limits are enabled, and a user inputs a scheduled report request that exceeds his maximum allotted number, he will receive a message as shown below. He will then have to review his existing report requests, and delete one of them before he can submit a new request.



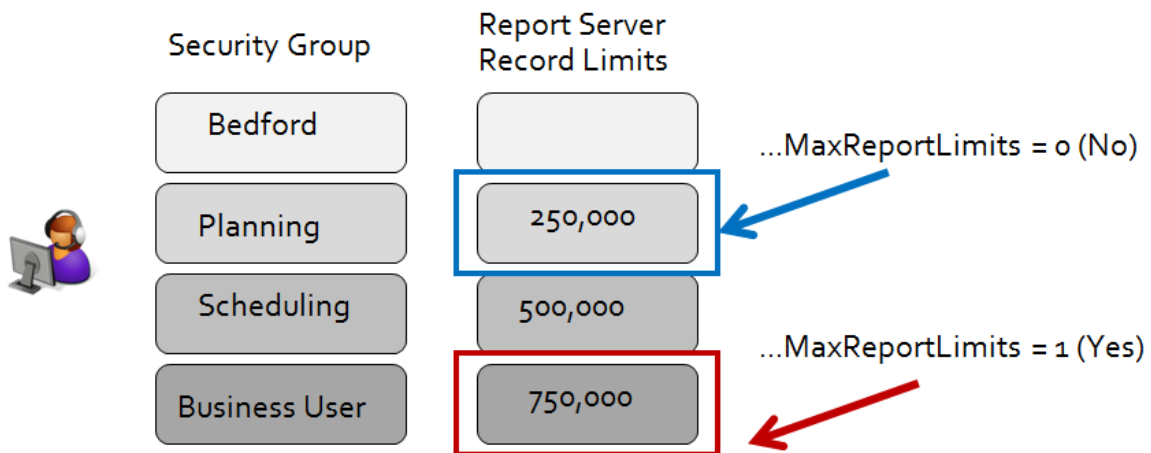
4.5 Minimum versus Maximum Value of Security Group limits

Your users may apply to multiple security groups. Within those different security groups, you may or may not set the variety of report limits which are available. These include Ad Hoc Preview limits, Report Server Record Limits and Scheduled report limits.

If a user is a member of multiple security groups, you may want the minimum or maximum value of the record limit to be applied to the user. In Maximo 76, a new property setting, `mxe.report.MaxReportLimits`, determines enables you to configure this value.

As shown in the example below, the user is a member of multiple security groups: Bedford, Planning, Scheduling and Business User. Two of those security groups have report server record limits set.

- If you define the property value, `mxe.report.MaxReportLimits` to be 0 or no, the minimum limit would be applied to the user which in this case is 250,000 server record limits.
- If you define the property value, `mxe.report.MaxReportLimits` to be 1 or yes, the maximum limit would be applied to the user which in this case is 750,000 server record limits.



4.6 Administration: Queuing

When numerous reports execute at once, performance can be adversely affected. This performance impact can be seen within applications, or in the amount of time it takes to process these reports. Because of potential performance concerns, Maximo provides functionality to limit the number of jobs that can be processed at any given time. This distributes the load on the server from reports, and reduces the wait time for report users.

Report Queuing manages the load of report jobs. It is made up of two components: scheduled report jobs and immediate reports jobs.

- Scheduled report jobs are controlled by the Queue Manager. It manages the scheduled jobs that enter the queue and the workers who process the scheduled jobs.
- Immediate report jobs are not managed by the Queue Manager.

The Report Queuing process can be described with the diagram below. It highlights the property setting, `maxconcurrentrun`, which limits the number of report jobs that can be executed on a server at any time. The default value of this property value is 5.

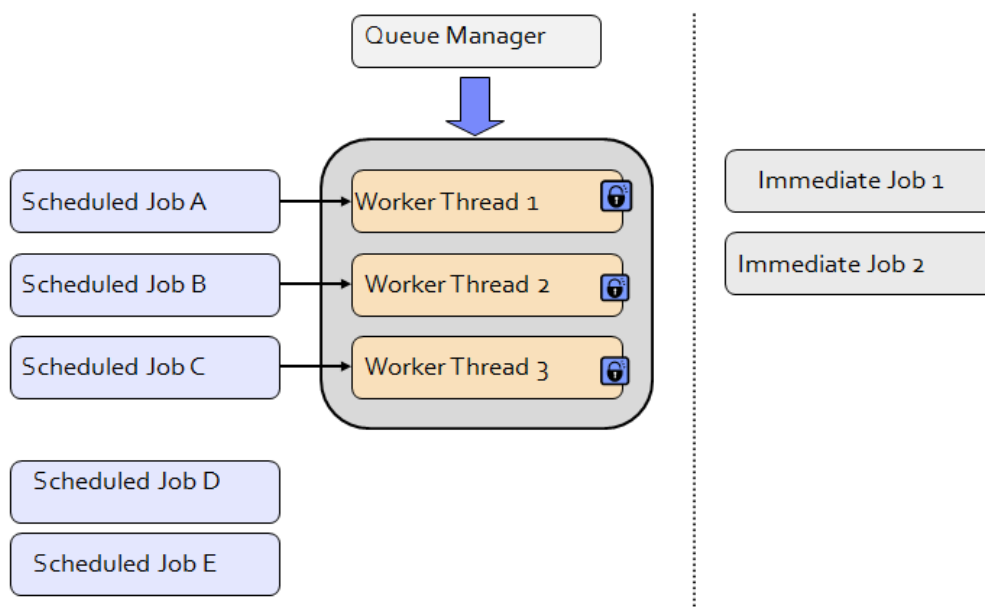
In this diagram, 2 immediate jobs are being processed, along with 3 scheduled report jobs. These 3 scheduled report jobs (A, B and C) are being controlled by the Queue Manager, and are being processed by 3 worker threads. The 2 immediate jobs are being processed by immediate job threads.

This is the maximum number of report jobs that can be processed as determined by `maxconcurrentrun`.

$$\# \text{ Actively Running Scheduled Jobs} + \# \text{ Immediate Jobs} \leq \text{maxconcurrentrun}$$

In the diagram below, this is:

$$3 \text{ (Actively Running Scheduled Jobs)} + 2 \text{ (Immediate Jobs)} \leq 5 \text{ (maxconcurrentrun value)}$$



Additionally:

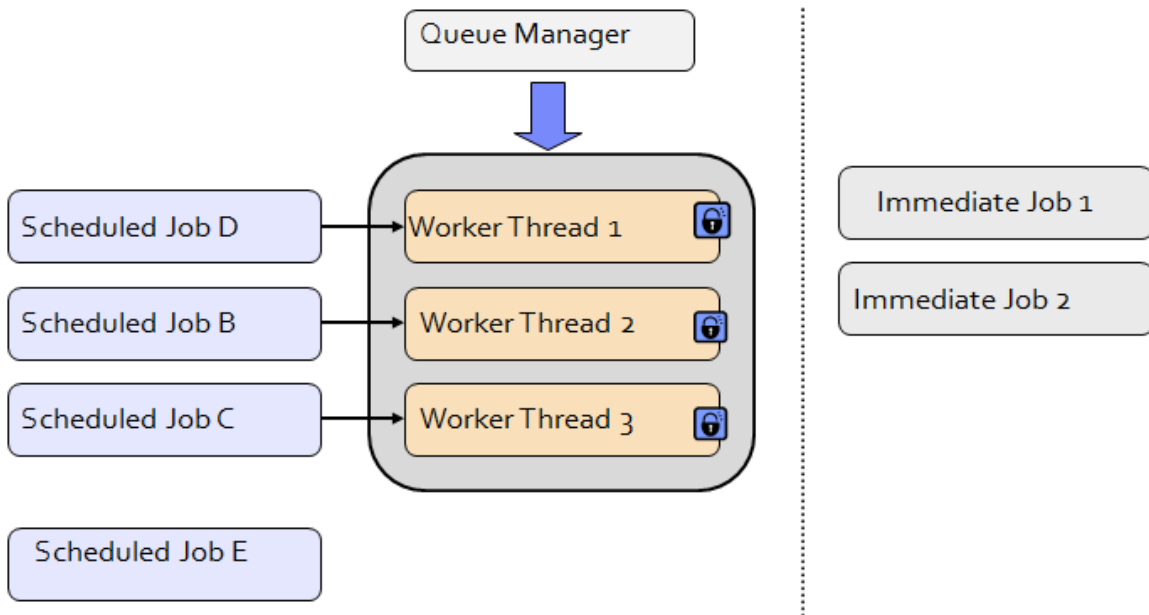
- (1) When it is time for a scheduled report to be run, the scheduled report job enters the Queue.

Scheduled Jobs D and E have entered the reporting queue, but they cannot be processed because there is no availability. (No available worker threads.)

- (2) The Queue Manager checks the queue for scheduled jobs using the property setting, `queueidletimeseconds`.
 - i. If jobs are present, then one job at a time is given to an available worker thread.
 - ii. Scheduled Jobs are submitted to the Queue Manager based on
 1. Ascending Order of Report priorities from Levels 1 to 5
 2. Ascending Order of Scheduled Job Run Time
 - iii. If there are no available worker threads, then the scheduled job will remain in the queue until a worker thread is available

Therefore, scheduled Jobs D and E will remain in the reporting queue until either one of the immediate or one of the scheduled report jobs completes.

This is shown in the updated diagram below where scheduled Job A has completed. The Queue Manager then moves scheduled Job D to worker thread 1, locks it and begins processing that D's report job request.

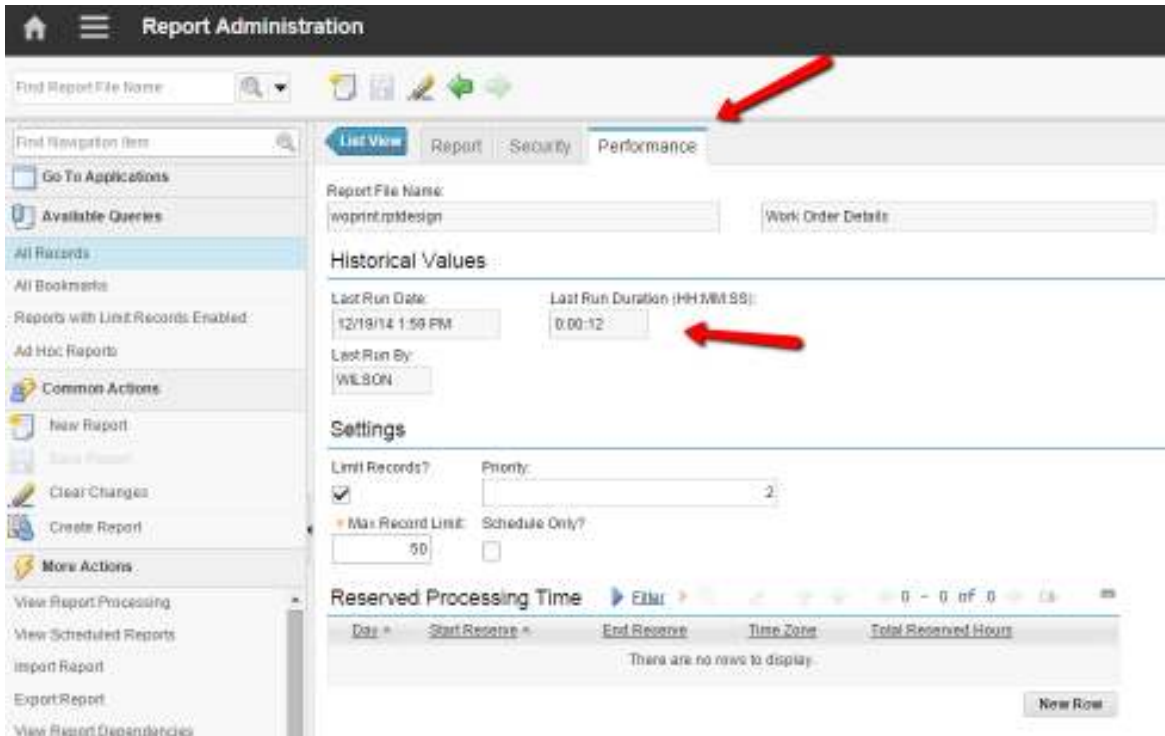


Notes

1. More information on the Property Files, Cron Tasks and Business Rules for Report Queuing is contained in the Maximo Reporting Features Guide referenced at the end of this guide.
2. Also, for Report Only Server Configurations, please review the 'Configuration Recommendations and Notes' Section in this guide for `maxconcurrentrun` settings for this configuration option.

4.7 Administration: Performance Tab

A Performance tab is available within the Report administrator application to quickly monitor performance on individual reports.



The screenshot shows the 'Report Administration' application interface. The 'Performance' tab is selected, indicated by a red arrow. The interface includes a search bar for 'Find Report File Name' and 'Find Navigation Item'. The 'Report File Name' is 'woprntptdesign', and the 'Work Order Details' is visible. The 'Historical Values' section shows 'Last Run Date' as 12/19/14 1:59 PM and 'Last Run Duration (HH:MM:SS)' as 0:00:12, with a red arrow pointing to the duration. The 'Settings' section includes 'Limit Records?' (checked), 'Priority' (2), 'Max Record Limit' (50), and 'Schedule Only?' (unchecked). The 'Reserved Processing Time' section shows a table with columns 'Date', 'Start Reserve', 'End Reserve', 'Time Zone', and 'Total Reserved Hours', and a message 'There are no rows to display'.

The first section of the Performance Tab is historical values. This contains value information from the last time the report was executed. In the example below, it shows that the 'Work Order Details' report was last executed by user Wilson on 12/19/14 and in 12 seconds.

This information is held in the REPORTUSAGELOG table and can also be displayed by executing the Report Usage Report.

Enabling the display of these values quickly gives report administrators a data point on how long reports are taking to execute. This is valuable information in a development-type environment as administrators can identify long-executing reports, and enable features like 'Schedule Only' or 'Record Limits' to minimize their performance impacts.

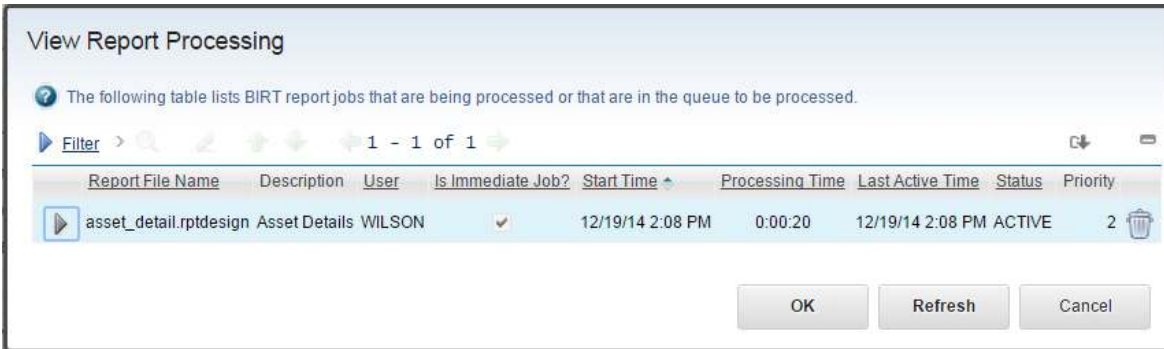
The second section of the Performance Tab is Settings. This contains the settings that directly impact performance. These settings are also displayed on the Report Tab, but are also enabled here for ease of use. For example, if the administrator sees a long executing report, he can quickly set its 'Limit Records?' Flag here on this tab.

The third and final section of the Performance Tab is Reserved Processing. This functionality is enabled for 'Schedule Only' Reports. More details on this can be found at the end of this document.

4.8 Administration: View Report Processing

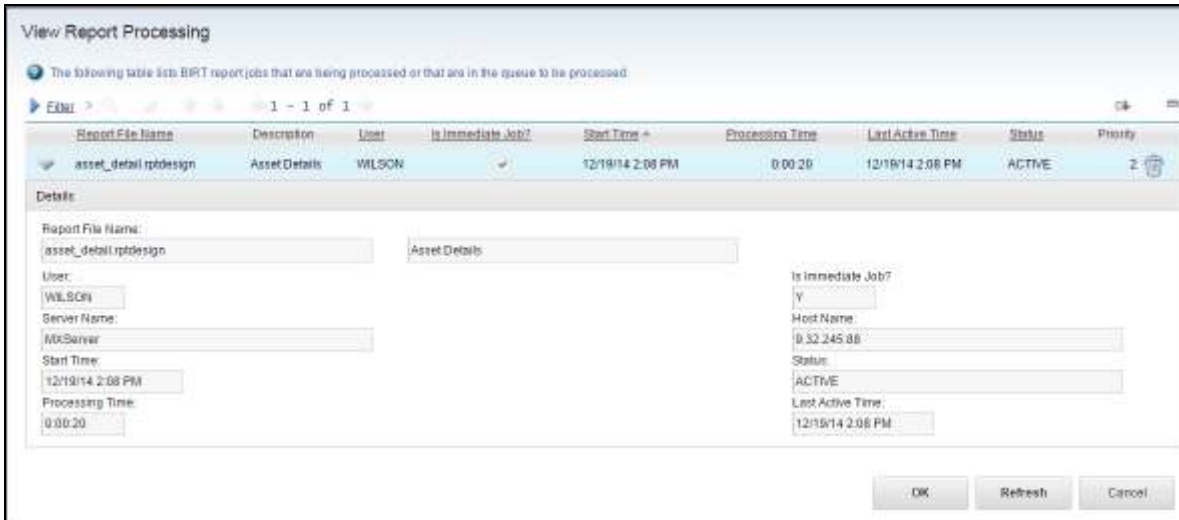
From within Report Administration, you can view which reports being executed. The View Report Processing dialog shows all reports that are either being processed by the report engine, or are in the queue waiting to be processed. The records are displayed based on ascending order of Start Time. This display of records enables the administrator to focus on those report jobs that are taking the longest time to execute.

In Maximo 76, a new field 'Last Active Time' was added to this dialog. This highlights if the report is still active, or if it may be stuck somewhere in its processing cycle.



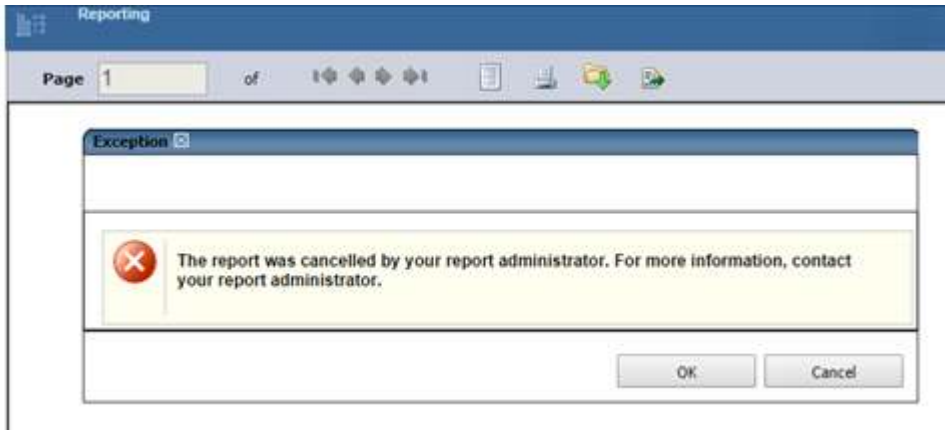
Clicking on the job details shows additional critical information. The server value is important as the report job may be processed on a different server.

For example, if a Cron Cluster is enabled for configured for report scheduling, then scheduled reports may execute on a different server.



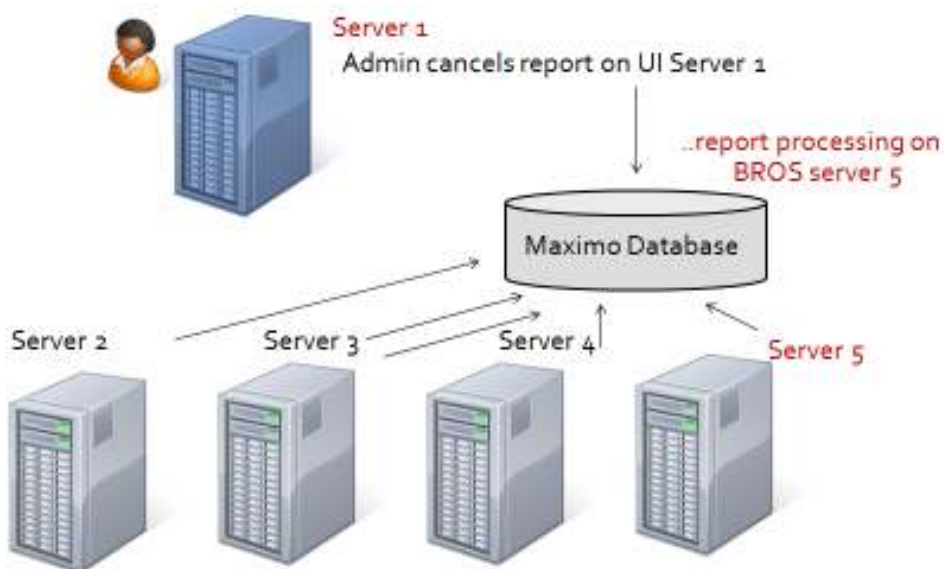
4.8.1 Administration: Report Cancellation

After viewing the report jobs that are processing, the Administrator can cancel a long running report job by selecting the Garbage Can. If it is an immediate job, the user will receive a message in the report viewer that it has been cancelled as shown here.



If the report job was scheduled, the user will receive his scheduled email with a notification that the report was cancelled by the administrator.

As noted throughout this guide, a report may be processing on a different server. In environments where multiple servers are configured, a new property setting has been added in Maximo 7.6. This setting, `mxe.report.birt.cancelreportinterval`, is the frequency that server checks database to see if reports canceled. This is highlighted in the diagram below where the administrator cancel s a report on Server 1, but the report is processing on Server 5. The property setting defines how often each server checks the database to see if the report is cancelled.



4.8.2 Administration: Managing Report Cancellation

You can analyze which reports have been cancelled via two new fields added in the Maximo 76 release in the REPORTUSAGELOG object. These fields are

ULSTATUS: Status of report job - which in this case would be cancelled.

ULMSG: Log message

These fields can be viewed when running the Report Usage report from Report Administration as highlighted below.

Start Date	End Date	Run Time (HH:MM:SS)	User	Application	Scheduled?	Server	Success?	Cancel?	Details
11/4/14 11:55:24 AM	11/4/14 11:55:44 AM	00:04:20	WILSON	NOTRACK	N	MUServer-SQS-CTrySUS	N	Y	Report is cancelled by Admin

Additionally, from within Report Administration, you could create your own ad hoc report on REPORT and REPORTUSAGELOG, if you want to narrow your results to only show for example the Report Status, along with userid, report name, and dates. An example of this is shown here.

Report File Name	Application	Start Date and Time	End Date and Time	Cancelled Job	User Id	Report Status	Log Message
rptUsage	NOTRACK	11/4/14 11:55:24 AM	11/4/14 11:55:44 AM	Y	WILSON	CANCEL_GENERIC	Report is cancelled by Admin

4.9 Administration: Report Escalations

The View Report Processing dialog is an excellent mechanism for your administrator to see which reports are processing. However, administrators may not frequently have the ability to monitor this dialog continuously for long running reports.

Beginning in Maximo 7.6, new features have been added which enable the administrator to be automatically notified of either inactive or long running reports. This new feature enables him to take action on these reports before negative performance impacts result.

Utilizing the REPORTLONG escalation, you can configure two escalation points - Start Time and Last Active Time.

- Start time is used to identify long running reports.

- Last active time is used to identify reports which do not have a recent database fetch. In these cases, the report may be stuck in a processing stage and needs to be cancelled.

If a report exceeds either of these limits, an email notification is sent to the configured administrator. This person is identified by the role, RPTADMIN.

With this dynamic notification, the administrator can view the details of the report in the View Report Processing dialog and take action.

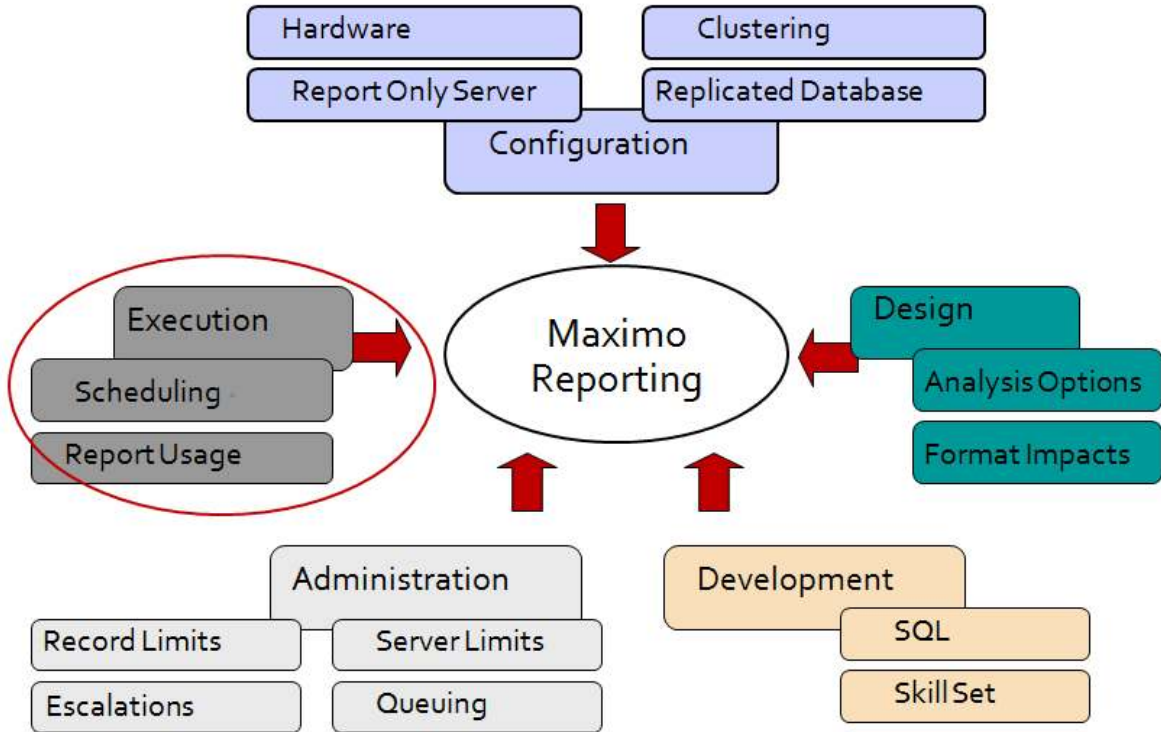
The screenshot displays the 'Escalations' configuration page in Maximo. The interface includes a left-hand navigation pane with options like 'Go To Applications', 'Available Queries', and 'New Escalation'. The main area shows the configuration for an escalation named 'Long Running Report Escalation'. Key fields include 'REPORTLONG' for the escalation name, 'REPORTJOB' for the job it applies to, and a 'Condition' field. The 'Escalation Points' section is expanded, showing two points: '1 STARTTIME' with a 10:00 minute interval, and '2 LASTACTIVETIME' also with a 10:00 minute interval. The 'Validation Results' section is currently empty.

Escalation Point	Elapsed Time Attribute	Elapsed Time Interval	Interval Unit of Measure	Organization	Calendar
1	STARTTIME	10:00	MINUTES		
2	LASTACTIVETIME	10:00	MINUTES		

5 Execution

How users execute reports has a significant impact on report performance. If multiple very large, complex reports are run during peak business hours, you could see performance delays in your environment.

To minimize these potential impacts, review the various ways in which reports can be run in this section.

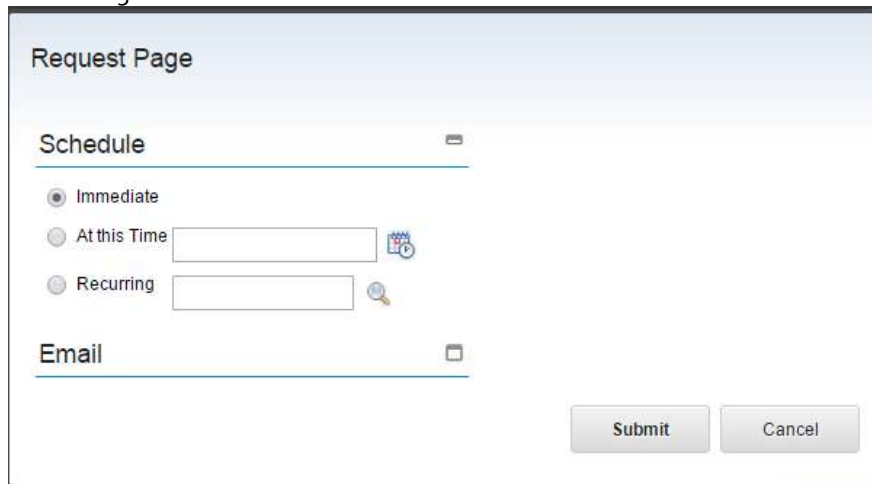


5.1 Execution - Scheduling

In Maximo, reports can be executed a variety of ways, including

- (1) Immediately
- (2) At a single selected date/time in the future
- (3) On a recurring basis

Immediate report jobs are the default Schedule option. These report jobs pass the user's report requirements to the report engine as soon as the user selects the 'Submit button'. The other two options – running a report at a single point in the future – or on a future recurring basis – involve scheduling. Report scheduling is enabled through a Cron Task.



The screenshot shows a web form titled "Request Page". It has two main sections: "Schedule" and "Email". The "Schedule" section has three radio button options: "Immediate" (which is selected), "At this Time" (with a date/time input field and a calendar icon), and "Recurring" (with a text input field and a clock icon). The "Email" section has a checkbox. At the bottom right of the form are two buttons: "Submit" and "Cancel".

Report Scheduling can be used as a performance enhancer by:

- A. Off-loading the report request to execute during Off-Peak Hours. Off-Peak Hours are defined as those hours in your individual business environment where the number of users actively using the system is minimized.

For example, scheduling a complex, analysis report like the Maintenance Cost Rollup to execute every Sunday at 4:00 AM may minimize the application and database performance impact rather than have the same report execute every Tuesday at 1:00 PM.

Note: The Schedule Report Only – Reserved Processing Times functionality can be used to enforce the use of these off-peak hours.

- B. Enabling the use of a Clustered cron server. By scheduling reports, the Clustered cron server can be used to execute the scheduled report versus the UI Server which is optimized to handle transactional requests and reports.

Executing the scheduling of reports is enabled thru a Cron Task Scheduler. The cron task is called REPORTSCHEDULE, and it references the class file that initiates it.

Note:

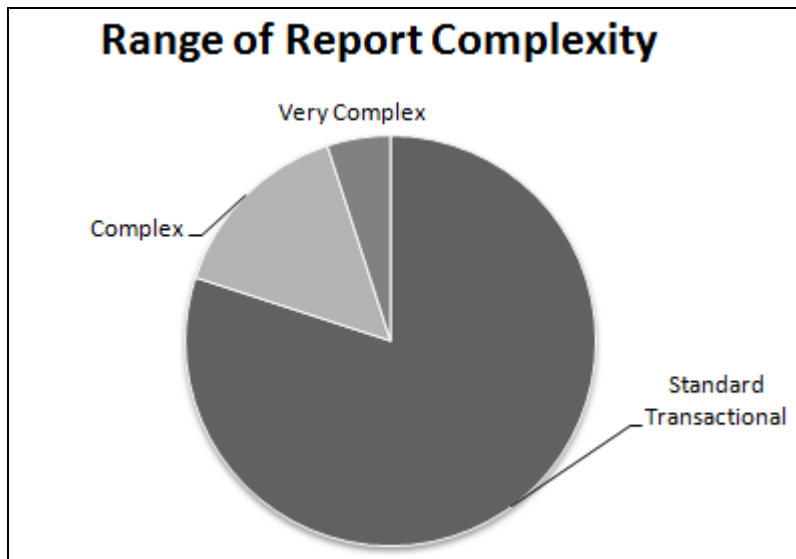
More information on Report Scheduling, including the Report Schedule Cron Task, the User's Scheduling Status and the Administration's View Schedule Report Action are contained in the Maximo Report Feature Guides referenced at the end of this document.

5.2 Execution: Schedule Only Reports

You can configure complex, batch reports to only be executed via report scheduling. This can prevent users from running large, complex reports during peak use of the application, database and report server, which can negatively impact overall system performance.

Before this functionality is described, let's review how this could be applied. In your environment, there is likely a wide range of complexity in reports. Some reports are very simple, like list reports or single grouped reports, whereas others are very complex due to the number of subreports they encompass or the hierarchy levels they span through. This Report complexity is defined by the processing the report has to do – not by the number of records it displays. So, for example, a 50 page list report could execute 10 times faster than a complex 2 page report due to the processing defined within the report's design.

The pie chart below represents a sample collections of reports. Standard, transactional reports are displayed in blue, with complex reports in red, and the most complex reports in yellow. These complex reports are time consuming to design and develop, and are continually analyzed for performance efficiencies. These are often referred to as batch reports, as they traditionally execute overnight due to their large processing requirements.



Because of the processing load complex reports have on a system, they can now be configured to only be executed via report schedules. Additionally, the very complex reports can be configured to only execute via schedules at specific times of the day. This will minimize the impact these complex reports have on overall system performance.

5.3 Execution - Schedule Only Reports – Reserved Processing Time

In addition to enabling Schedule Only functionality for very complex, batch reports, you can also configure what times are available for the report scheduling. This will prevent users from scheduling report jobs of complex reports at the peak times of server use. For example, you may not want your users to run very complex batch reports on Monday morning, when many users are trying to execute their daily or weekly transactional reports.

Reserved Processing time will detail the busy days and time within a week that server processing time is reserved for immediate report jobs and other maximo application functionality. This functionality will be based on days of the week. It is not based on calendar days, example January 1, due to fluctuations with calendars and holidays.

Reserved Processing Time can be set on any day of the week (Monday thru Sunday) and for any amount of time less than 24 hours. On the Performance Tab in Report Administration, the administrator will click New Row and specify the day of the week, and the times during that day that report processing is NOT available for that report job. This functionality is enabled by individual report because of the range of report complexity.

When the administrator clicks new row, multiple fields will display, including Day, Start Reserve, End Reserve and Total Reserved Time (Hrs).

The screenshot shows the 'Performance' tab in the Report Administration interface. At the top, there are tabs for 'List View', 'Report', 'Security', and 'Performance'. Below the tabs, the 'Report File Name' is 'maintaincost_by_system.rptdesign' and the 'Cost by System' is 'Cost by System'. The 'Historical Values' section shows the 'Last Run Date' as '12/19/14 4:27 PM', 'Last Run Duration (HH:MM:SS)' as '0:00:02', and 'Last Run By' as 'WILSON'. The 'Settings' section includes 'Limit Records?' (unchecked), 'Priority:' (empty), 'Max Record Limit' (empty), and 'Schedule Only?' (checked). The 'Reserved Processing Time' section features a table with columns for Day, Start Reserve, End Reserve, Time Zone, and Total Reserved Hours. The table contains two rows: MONDAY and TUESDAY, both with a start reserve of 7:00 AM, an end reserve of 3:00 PM, and a total reserved time of 8:00 hours. The table also includes a 'Filter' button and pagination information '1 - 2 of 2'.

Day	Start Reserve	End Reserve	Time Zone	Total Reserved Hours
MONDAY	7:00 AM	3:00 PM	America/Chic	8:00
TUESDAY	7:00 AM	3:00 PM	America/Chic	8:00

What the Schedule Only Report Looks like to End User

If the 'Schedule Only' field is enabled for a report, the report cannot be run immediately. The user will only have the choice to run this report either at a single point in the future, or on a recurring basis.

To highlight this functionality to the user

- A. The Immediate Runtime Option is hidden under the Scheduling Section
- B. The 'At this Time' Schedule Option will become the default.

The Schedule availability shown on the report's request page will show the opposite of what is shown in Report Administration. In Report Administration, the administrator is concerned with reserving time that the report should NOT be enabled to execute. Hence, it is called reserved processing time. However, for a report user, he wants to know when he can execute a report. He doesn't want to know when he can't execute a report – hence, the values will be shown as Schedule Availability.

The Report Availability also displays the schedule time availability time in the Time Zone in which it was SET. This may not match the time zone of the user.

Note: This functionality is expected to be enabled on complex reports executed by fairly technical users, who will understand time zones and their conversions.

Notes:

1. To enable the Schedule Availability Section, the database table, REPORTPROCAVAIL, is used. This table holds the opposite of the REPORTPROCRESERVE, which holds the report's reserved processing time. REPORTPROCAVAIL is required so end users can filter/search on Report Processing Availability from the Report's Request Page.
2. Due to potential conflicts with varying time zones - and the complexity involved with Daylight Savings time being utilized in some time zones, but not others – the administrator can only reserve processing time for each report using the same time zone.

This means that if he reserves a processing time of 7 AM to 4 PM EST for the Asset Cost Rollup Report, any other reserved processing time made for Asset Cost Rollup must be made in the EST Time Zone.

3. The Schedule Only? Flag must be enabled for an administrator to set Reserved Processing Time.

5.4 Execution - Report Usage

Report usages is critical to help you answer important questions often raised by Development/IT including:

- Which reports take the longest to execute?
- Does anyone really use this report? And if so, who are they?
- Are users taking advantage of the scheduling functionality? Or are they causing performance constraints by always running reports immediately?

The Report Usage report details the information collected from the Report Usage Table. This report is available from the Report Administration application. The report contains information on reports that are executed immediately, scheduled to execute at a future time, or are accessed via hyperlinks.

One of the most important fields in the Report Usage report is the Run Time Field. The data from this field comes from REPORTUSAGELOG.RUNTIME, which is stored in the database in milliseconds. The report converts the data from milliseconds to HH:MM:SS Format for ease of analysis.

Report Usage										
Report Name: Asset Details				File Name: asset_details.rpt						
Application: ASSET										
Start Date	End Date	Run Time (HH:MM:SS)	User	Application	Scheduled?	Email File Type	Server	Success?	Cancel?	Details
12/18/14 02:58:26 PM	12/18/14 02:08:47 PM	00:01:21	WILSON	ASSET	N		MSDServer	Y	N	
Report Name: Cost by Location				File Name: cost_location.rpt						
Application: LOCATION										
Start Date	End Date	Run Time (HH:MM:SS)	User	Application	Scheduled?	Email File Type	Server	Success?	Cancel?	Details
12/18/14 01:19:51 PM	12/18/14 01:09:27 PM	00:08:36	WILSON	LOCATION	N		MSDServer	Y	N	
Report Name: Cost by System				File Name: cost_system.rpt						
Application: LOCATION										
Start Date	End Date	Run Time (HH:MM:SS)	User	Application	Scheduled?	Email File Type	Server	Success?	Cancel?	Details
12/18/14 04:27:36 PM	12/18/14 04:27:32 PM	00:00:02	WILSON	LOCATION	N		MSDServer	Y	N	

Query Based Reports (QBRs) are also captured in the Report Usage Table. Because these reports often may be executed only a single time, their corresponding run times will be displayed in a separate section, called 'Transient Reports' at the end of this Report.

Report Usage									
Transient Reports									
Start Date	End Date	Run Time (HH:MM:SS)	User	Application	File Name	Server	Success?	Cancel?	Details
12/18/14 04:58:49 PM	12/18/14 04:58:51 PM	00:00:02	WILSON	ASSET	ASSET_WILSON_141800372.rpt	MSDServer	Y	N	
12/18/14 04:59:24 PM	12/18/14 04:59:25 PM	00:00:01	WILSON	REPORT	REPORT_WILSON_1418005346.rpt	MSDServer	Y	N	

6 Additional References

The following lists additional references available at the time this document was prepared.

Maximo BI Wiki Site

Dynamic location to access the latest information on reporting.

<http://ibm.co/13V4RZZ>

or

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Maximo%20Asset%20Management/page/Reporting>

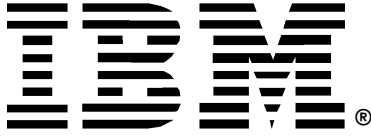
BI Documentation – Listing of all reporting documentation by release and feature

This includes the documentation referenced throughout this guide including Report feature guides, development guides and report booklets.

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Maximo%20Asset%20Management/page/Reporting%20Documentation>

BI Demos – Listing of recorded demos by release and feature

<https://www.ibm.com/developerworks/community/wikis/home?lang=en#!/wiki/IBM%20Maximo%20Asset%20Management/page/Demo%20Recordings>



© Copyright IBM Corporation 2015
IBM United States of America
Produced in the United States of America
US Government Users Restricted Rights - Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

*IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785
U.S.A.*

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PAPER "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes may be made periodically to the information herein; these changes may be incorporated in subsequent versions of the paper. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this paper at any time without notice.

Any references in this document to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
4205 South Miami Boulevard
Research Triangle Park, NC 27709 U.S.A.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information is for planning purposes only. The information herein is subject to change before the products described become available.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at <http://www.ibm.com/legal/copytrade.shtml>.

Adobe, the Adobe logo, PostScript, and the PostScript logo are either registered trademarks or trademarks of Adobe Systems Incorporated in the United States, and/or other countries.

IT Infrastructure Library is a registered trademark of the Central Computer and Telecommunications Agency which is now part of the Office of Government Commerce.

Intel, Intel logo, Intel Inside, Intel Inside logo, Intel Centrino, Intel Centrino logo, Celeron, Intel Xeon, Intel SpeedStep, Itanium, and Pentium are trademarks or registered trademarks of Intel Corporation or its subsidiaries in the United States and other countries.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

ITIL is a registered trademark, and a registered community trademark of the Office of Government Commerce, and is registered in the U.S. Patent and Trademark Office.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Cell Broadband Engine is a trademark of Sony Computer Entertainment, Inc. in the United States, other countries, or both and is used under license therefrom.

Linear Tape-Open, LTO, the LTO Logo, Ultrium, and the Ultrium logo are trademarks of HP, IBM Corp. and Quantum in the U.S. and other countries.

Other company, product, or service names may be trademarks or service marks of others.