IBM Cognos Proven Practices - System Management Methodology for IBM Cognos 10: System Metrics

Nature of Document: Tip or Technique; Product(s): IBM Cognos Business Intelligence v10; Area of Interest: Infrastructure

Business Analytics Proven Practices Team
Business Analytics Proven Practices Team
IBM

03 December 2014
(First published 17 October 2011)

This is part of a group of documents and examples commonly referred to as the System Management Methodology (SMM). The SMM provides information and techniques as examples of how an administrator uses the standard features of IBM Cognos Administration along with IBM Cognos Business Intelligence functionality in order to increase their own productivity and pro-actively manage IBM Cognos BI applications, users, and servers.

View more content in this series

Introduction

Purpose

This is part of a group of documents and examples commonly referred to as the System Management Methodology (SMM). The SMM provides information and techniques as examples of how an administrator uses the standard features of IBM Cognos Administration along with IBM Cognos Business Intelligence functionality in order to increase their own productivity and pro-actively manage IBM Cognos BI applications, users, and servers. By combining the live view of IBM Cognos BI system activity provided by the IBM Cognos Administration features with system trending information using the reporting and analytical features of IBM Cognos BI, administrators can get a full view of BI system utilization.

Additional information about the Administrative features of IBM Cognos BI are available within the IBM Cognos Business Intelligence Administration and Security Guide.

Applicability

The documented information and technique(s) apply to all IBM Cognos Business Intelligence 10.1.1, 10.2.1 and 10.2.2 installations. Although precautions are taken to ensure that the
information and technique(s) span newer releases, some of the content may become obsolete and/or no longer applicable.

Content found in an appendix pertains to previously released System Management Methodology topics/techniques that have been replaced with a newer technique. The newer technique(s) will appear in the regular chapters of the document. When a new topic/technique is added that replaces a technique, it will be highlighted in the document.

**Exclusions and Exceptions**

The scope of the documents in this series may not include detailed steps on how to use the product as this information is contained within the core product documentation. For example, the document “System Management Methodology For IBM Cognos 10 – Setting Thresholds” deals with thresholds that can be applied to the system metrics. The document offers guidance on how to interpret those metrics to apply default thresholds programmatically but does not include the actual steps required to manually set or modify a threshold.

**Introduction to System Metrics**

With each new release of IBM Cognos BI since version 8.3, more and more features are added to support the management of the IBM Cognos server platform availability, users, applications, scheduled tasks, and data sources. Each service, dispatcher, server and server group exposes relevant metrics that provide administrators with better insight into the status and overall health of the various components that make up an IBM Cognos Business Intelligence installation.

The metric values are live (real time) and reside in an MBean in the Java environment. Because the metrics are live, they are dynamic and will only be reset when the IBM Cognos Business Intelligence service/process is restarted or when an explicit request, either manual or programmatic, is executed. More detail on this topic can be found in **Services Section**.

An important note regarding the metrics, and the administration console in general, is that there is no auto-refresh feature. This was a design decision based on the general feedback from administrators, that this limited the ability to thoroughly analyze a series of metrics if the values were changing on a regular basis. There are a few manual refresh options available from within the console:

- **Fragment refresh** - This toolbar reset option will refresh the values of the contents of the window attached to the toolbar. For example, refreshing the Metrics frame will update the metric values but will not change the contextual item or refresh any of the values in any of the other frames.
- **Page refresh** - This button is available in the main IBM Cognos Business Intelligence toolbar located at the top of the browser page. When pressed, the entire page being viewed will be refreshed. For example, pressing this button when viewing the System task will refresh the Scorecard, Metrics and Settings frames without losing context.
- **Browser refresh** – Using the browser refresh button to update the administration console will cause all pages to be refreshed. This action will cause all context to be lost and once
the refresh has occurred, the default view allowed by your administrative capabilities will be displayed.

To provide reference as to the timeliness of the information being viewed, there is a summary bar at the bottom of the frame that displays the last time that a refresh occurred.

**Figure 1. The frame status summary bar showing the time the frame was last refreshed**

| Last refresh time: September 6, 2011 4:25:56 PM |

**System Task**

IBM Cognos Business Intelligence system metrics are displayed as part of System task in the Administration console. To view the metrics,

- From within IBM Cognos Connection, click **Launch > IBM Cognos Administration** to open the IBM Cognos Administration console.
- Click the **Status** tab and then click the **System** task.

The system task interface is divided into three fragments.

- **Scorecard** – Left hand frame that displays a summary of the overall health of the components that make up the environment.
- **Metrics** – Upper right hand frame that lists all of the metrics, and their score, that pertains to the object in focus (from the scorecard fragment). The default is to display the metrics for the overall system (environment).
- **Settings** – Lower right hand frame that shows a read-only view of the configuration parameters that pertain to the object in focus.

**Standard Scorecard**

There are two views available with the scorecard fragment, the default view and the comparative view. The default view, or standard view, allows administrators to navigate through the system topology to verify the health of the servers, dispatchers and services as well as their corresponding metrics. The comparative view provides the ability to watch predefined key metrics as they relate to similar objects.

The **Scorecard** panel provides administrators with a view of the entire IBM Cognos Business Intelligence topology. The metric status indicator lights show the overall health of the object compared to thresholds threshold values that can be assigned to each metric by an Administrator. The status of the object is also displayed in the Status column. The Status is displayed using the following terms,

- Partially available – indicates that one or more of it’s children is unavailable (offline)
- Available – indicates that the object is online
• Unavailable – indicates that the object is offline

Figure 2. The Scorecard panel shows the names of the servers, the services available, the availability and status indicators

Object status is inherited up the object hierarchy. If one server in the object hierarchy is online and available, while another server is offline, thus unavailable, the overall system status becomes partially available due to the fact that at least one child is unavailable.

The underlying dispatcher(s) and services can be viewed by drilling down on the server name, which maintains the parent/child relationships. For example, drilling down on a server name would reveal the dispatchers running on that server.

The Scorecard panel display can also be filtered using the down arrow icon beside the All servers object. This drop down menu allows for filtering of all servers, server groups, dispatchers, services as well as being to select individual services.
For example, selecting the Report service would provide a list of all Report services, their overall health as well as their current state. The ability to perform metric comparisons though would not be possible unless you switched focus between Report services.

### Comparative Scorecard

As mentioned in the previous section, monitoring services side by side in the standard view does not provide the ability to easily compare metrics. There is however a comparative view that allows administrators to see certain metrics as they pertain to related objects in one single view. To access this view, maximize the scorecard frame by pressing the expand icon. This will maximize the frame within the console and provide the comparative view.

**Figure 4. The comparative scorecard view showing metrics for each ReportService instance within the BI system**
At first glance it is difficult to determine which dispatcher each one of the Report services belongs to. The parent dispatcher is visible by hovering over the Service icon which will produce a tool-tip with the parent’s name (for example, http://sottpfw:9300/p2pd).

**Metrics**

The traffic indicator lights mentioned in the previous sections are derived from values of metrics pertaining to the IBM Cognos BI application. These metrics in IBM Cognos Business Intelligence empower administrators to be able to effectively optimize and monitor their environment(s).

The Metrics fragment, located in the upper right hand corner of the System task, displays the metrics that correspond to the object in the Scorecard frame that is in context. For example, when the System task is first opened the default view is of the entire topology or environment. This means that the default view of the Metrics fragment displays the metrics that pertain to the entire environment. Drilling down into the various components that make up the environment will also result in the changing of the Metrics view.

**Figure 5. The Metrics fragment from the IBM Cognos Administration interface showing all the metric groupings and threshold indications**

The image above displays the default view of the Metrics frame and all of the metric groupings that pertain to the system. The object in context is visible through the fragment title, in this case **Metrics - System**. As the context changes in the Scorecard fragment, the title of the Metrics frame will also change.

Just beneath the toolbar of the Metrics fragment there is a summary bar that indicates the total of amount of each threshold indication. This summary also acts as a means to filter the view to quickly monitor the threshold indications that are of importance. By default all indication types are selected, including the metrics that do not have a threshold assigned to them. The status of system metrics are represented by traffic light style indicators – a red square, an amber diamond, and a green dot are each used to represent the value of a metric compared to its defined threshold. Red indicates a bad status, amber indicates a transitioning status, and green indicates a good status.
Figure 6. Metrics fragment summary bar and status view filter

To quickly filter the list of metrics to only display the metrics that have a poor and average score, un-check the green (circle) indicator and the “No metric score” check boxes. By using these filters, only the metrics that may require attention are displayed.

Figure 7. A Metrics fragment view where only poor and average threshold indicators are selected

Settings

The last frame in the lower right hand corner of the System task is the Settings fragment. This read-only window displays all of the configuration parameters, and their values, for the object that is currently in focus in the Scorecard frame. This helps to provide administrators with some contextual information regarding the metric values that are being monitored.

As mentioned, this is a read-only display but a dialog to change the values can be invoked by clicking the Set Properties button. The resulting dialog box will present all of the configuration parameters that can be changed for the current view. This eliminates the need to have to navigate through numerous levels of dialogs to change the settings.

Individual Metrics

The system metrics are broken into three dimensions: individual metric, a metric group and the service to which they pertain. The complete list of metrics would contain metrics that have numerical values as well as timestamps, but this section focuses on the most of the individual metrics that have numerical values and what they represent.

AverageTimeInQueue

The average time in queue is calculated based on the total amount of time that all requests have spent in the queue divided by the total number of requests that have been in the queue. This averaged value maps to the ‘latency’ metric in the administration console.

FailedRequestPercent

This metric displays the percentage of failed requests that have occurred. The calculation is simple, \((\text{number of failed requests} / \text{number of received requests}) \times 100\).

The percentage metrics (failed and successful) are excellent metrics to monitor when resetting the metrics is either not possible or is not going to be done. The reason for this is that thresholds built on the percentage metrics will always be in play regardless of when the last reset has occurred.
With count metrics for failed and successful requests (NumberOfFailedRequests and NumberOfSuccessfulRequests) thresholds will only occur after a period of time. For example, if the threshold score is set to turn red after 50 failed requests, once the threshold is exceeded (one day, one week, one month, etc) the threshold score will always be red until the service is restarted or the metrics are reset.

The percentage metrics will change over time. Using the previous example, if the failed requests hit 50 after the first 50 requests, the value would be 100% and more than likely would result in a threshold score of red. From that point forward, if every request was successful, the metric value would decrease thus moving the red score to yellow and then eventually green. Due to this, if only the percentage metrics were being monitored via thresholds, no resetting of metrics would ever need to occur.

**LastResponseTime**
This metric displays the duration of the most recently received request, either failed or successful.

**MillisecondsPerSuccessfulRequest**
This is the average amount of time spent processing a successful request.

**NumberOfConfiguredProcesses**
The number of processes that have been configured as part of the properties page(s) for the object in context.

**NumberOfFailedRequests**
Number of failed requests, not to be confused with the failed request percentage metric, is a cumulative count of the amount of failed requests that have occurred since the last reset.

**NumberOfProcessedRequests**
Specifies the amount of received requests that have been processed by the dispatcher.

**NumberOfProcesses**
The value for this metric is an indication of what the current amount of active processes is been at the time the metric is being viewed or was exported.

**NumberOfProcessesHighWaterMark**
The value for this metric is an indication of what the highest amount of active processes has been since the metric was last reset.

**NumberOfProcessesLowWaterMark**
The value for this metric is an indication of what the lowest amount of active processes has been since the metric was last reset.

**NumberOfReceivedRequests**
The amount of total requests that have been received.

**NumberOfRequests**
Specifies the amount of requests that have passed through the queue since the last time the metrics were reset. This value maps to the ‘number of queue requests’ in the administration console.

**NumberOfSessions**
Indicates the amount of user sessions that are currently active in the environment.

**NumberOfSessionsHighWaterMark**
Indicates the maximum amount of user sessions that were active in the environment at one time.

**NumberOfSessionsLowWaterMark**
Indicates the minimum amount of user sessions that were active in the environment at one time. Based on environment traffic, this number should more than likely be zero.

**NumberOfSuccessfulRequests**
Number of successful requests, not to be confused with the successful request percentage metric is a cumulative count of the amount of successful requests that have occurred since the last reset.

**QueueLength**
The amount of requests currently in the queue. This is a metric that can not be assigned a threshold.

**QueueLengthHighWaterMark**
The value for this metric is an indication of what the highest amount of requests in the queue has been since the metric was last reset.

**QueueLengthLowWaterMark**
The value for this metric is an indication of what the lowest amount of requests in the queue has been since the metric was last reset. Typically this number will be zero if the metrics are reset due to a server restart as there would be no queued items at start up. If the metrics were reset manually through the product interface, this value would be greater than zero in a high volume application.

**ResponseTimeHighWaterMark**
The value for this metric shows the longest period of time spent processing a request, either successful or failed.

**ResponseTimeLowWaterMark**
The value for this metric shows the shortest period of time spent processing a request, either successful or failed.

**ServiceTimeAllRequests**
The service time metrics show the amount of time that was spent processing the requests. This particular metric value is the total amount of processing time that was used for all requests, including both failed and successful.
**ServiceTimeFailedRequests**
This particular metric value is the total amount of processing time that was used for all failed requests.

**ServiceTimeSuccessfulRequests**
This particular metric value is the total amount of processing time that was used for all successful requests.

**SuccessfulRequestPercent**
This metric displays the percentage of successful requests that have occurred. The calculation is simple, \((\text{number of successful requests} / \text{number of received requests}) \times 100\).

**SuccessfulRequestsPerMinute**
The definition of this metric differs slightly from the traditional definition, or perception, of successful requests per minute. The value does not indicate an ongoing average from minute to minute it is an indication of how many requests have been processed during the amount of time the system has spent processing them. The formula would be \((\text{number of successful requests} \times \text{service time for successful requests}) / 60 \text{ seconds}\).

For example, 10 requests are executed successfully and the server has spent 30 seconds executing the request. When looking at the metric after a minute, the traditional definition would indicate the average is 10 requests per minute. After the second minute, the value would be 5, etc. The actual use of this metric in IBM Cognos Business Intelligence would be 20 after one minute and would still be 20 after 2 minutes.

This algorithm shows what the average successful requests is based on the amount of processing time it took to execute them and not the actual time. This is done to provide a real value that isn't impacted by periods of inactivity. This metric is a great way to track server throughput.

**TimeInQueue**
This cumulative metric shows the total amount of time that has been spent by all objects in the queue. For example, if 30 requests have been in the queue at some point, each with a queue time of 1.5 seconds, the value for the metric would be 45 as the total time spent \((30 \times 1.5)\) is 45 seconds.

**TimeInQueueHighWaterMark**
Displays the longest amount of time that one object has spent in the queue.

**TimeInQueueLowWaterMark**
Displays the shortest amount of time that one object has spent in the queue.

Typically this number will be zero if the metrics are reset due to a server restart as there would be no items in the queued start up. If the metrics were reset manually through the product interface, this value would be greater than zero in a high volume application.

**Uptime**
The length of time that the Java Virtual Machine (JVM) has been running.

**Metric Groups**
The individual metrics are divided into three main metric groups.

- **Request**: These metrics pertain to the specific requests that are handled by each component in the environment. They include, but are not limited to, such metrics as the amount of processed requests, the percentages of successful versus failed requests, the amount of processing time for these requests.

- **Queue**: These metrics provide insight into the amount of requests that are not handled immediately and therefore placed into a queue to be processed when the resources become available. They include, but are not limited to, such metrics as the amount of requests that have been in the queue, the length of the queue as well as how much time requests have spent in the queue.

- **Process**: These metrics display information regarding the amount of processes required by the product to function. Metrics such as number of current processes as well as the maximum number of processes that were spawned are available.

There are a few metrics located outside of the three main metric groups (for example, JVM uptime and heap size information) but the majority of the individual metrics are a part of the three metric groups.

**Services**
The final dimension to the system metrics is how the individual metrics and metric groupings relate to the service to which they are associated. Understanding what actions are performed by each of the services provides greater insight into the values that are being reported. The services listed below are core IBM Cognos Business Intelligence services. Other services may be introduced to the environment as additional products are installed – for example the PowerPlay service when PowerPlay Studio is included in the product suite.

**Agent service**
The agent service is responsible for running agents. This service runs the conditions and creates and stores the generated event list. The service determines which tasks to execute and forwards those tasks to the monitor service for execution. In addition to running agents, the agent service also runs two other types of specialized tasks,

- Stored procedures using IBM Cognos Report Server
- Web service tasks

**Annotation service**
The annotation service enables the addition of commentary to reports using IBM Cognos Business Insight. These comments persist throughout different versions of the report.

**Batch report service**
The batch report service manages background requests to run reports and provides output on behalf of the monitor service.
**Content manager service**
The content manager service interacts with the content store. It performs object manipulation functions such as add, query, delete, update, move, and copy. It also handles the content store management functions, such as import and export.

**Content manager cache service**
The cache service enhances the overall system performance and IBM Cognos Content Manager scalability by caching frequent query results in each dispatcher.

**Delivery service**
As the name implies, the delivery service delivers content. Email, news items, and report output that is written to the file system are examples of content that is handled by the delivery service. Part of this service is a persistent email queue that is in place to guarantee that the items are forwarded to the configured SMTP server.

**Event management service**
The event management service is the service that handles scheduling. Part of the scheduling aspect is the control over cancelling, suspending, and releasing scheduled tasks. For tasks that are already entered the queue, requests to cancel, release, or suspend are forwarded from the event management service to the monitor service. The information found as part of the Upcoming Activities task in the IBM Cognos Administration console is also provided by the event management service.

**Graphics service**
The graphics service produces graphics on behalf of the report service. Graphics can be generated in the following formats:

- Raster
- Vector
- Microsoft Excel XML
- Portable Document Format (PDF)

**Human task service**
The human task service enables the creation and management of human tasks. A human task such as report approval can be assigned to individuals or groups on an ad hoc basis or by any of the other services.

**Index data service**
The index data service provides basic full-text functions for storage and retrieval of terms and indexed summary documents.

**Index search service**
The index search services provides search and drill-through functions, including lists of aliases and examples.

**Index update service**
The index update service provides write, update, delete, and administration functions.
Interactive Discovery Visualization service
This service provides content to IBM Cognos products to support interactive discovery and visualization functionality used by IBM Cognos Workspace.

Job service
Before jobs can be executed, they must first be prepared, meaning that the steps of a job must be analyzed for issues such as circular dependencies in nested jobs and resolution for run options that are part of the jobs. The job service completes these tasks and then sends the job to the monitor service for execution.

Log service
The log service creates log entries that are received from the dispatcher and other services. The log service is called regardless of which logging output is specified (for example database, file, remote log server, and so forth).

Metadata service
The metadata service provides support for data lineage information that is displayed in Cognos Viewer, IBM Cognos Report Studio, IBM Cognos Query Studio, and IBM Cognos Analysis Studio. Lineage information includes information such as data source and calculation expressions.

Metric studio service
The metric studio service provides the IBM Cognos Metric Studio user interface for the purposes of monitoring and entering performance information.

Mobile service
The mobile service provides the ability to send content to mobile devices, and handles requests from mobile devices.

Monitor service
The monitor service handles all of the requests set to run in the background, including scheduled tasks, reports that are set to run and then email the results, and jobs. Because the monitor service can receive more requests than can be executed, it also queues requests and waits for resources to become available for the required service. When a service indicates that there is sufficient bandwidth, the monitor service then forwards the task to the appropriate service for execution. Because the monitor service handles all of the background tasks, writing history information about the individual task executions is the responsibility of the monitor service.

The exceptions to this process are the history details for deployment and IBM Cognos Search indexing tasks, which are written directly to the content store using the IBM Cognos Content Manager component. The information found as part of the Current Activities task in the IBM Cognos Administration console is also provided by the monitor service.

Presentation service
The presentation service provides the display, navigation, and administration capabilities in IBM Cognos Connection. It also receives generic XML responses from other services and transforms them into output format, such as HTML or PDF. Another function of the presentation service is to
send the saved content when a request to view saved output is made. If a request to execute the report is made from inside of Cognos Viewer, the request is handled by the report service.

**Query service**
The query service manages Dynamic Query Mode requests and returns the result to the requesting batch report service or report service.

**Relational Metadata service**
The relational metadata service relies on the underlying relational database to return metadata required for reporting in Dynamic Query Mode.

**Report data service**
The report data service manages the transfer of report data between IBM Cognos BI and applications that consume report data, such as IBM Cognos Analysis for Excel, IBM Cognos Office Connection, and IBM Cognos Mobile.

**Report service**
The report service manages interactive report requests to run reports and provides the output for a user in IBM Cognos Business Insight or in one of the IBM Cognos studios.

**Repository service**
The repository service manages requests to retrieve archived report output from an archive repository. Unless you are using content archiving in 10.2 (which is similar to IBM FileNet functionality), you should not need this service enabled.

**System service**
The system service is used by the dispatcher to obtain application configuration parameters and provides methods for interfacing with locale strings that are supported by the application for support of multiple languages.

### Accessing System Metrics via JMX

IBM Cognos BI system metrics are maintained by the Cognos Dispatcher and stored within JMX MBeans. When system metrics values are viewed within the IBM Cognos Administration Console, the values presented within the Metrics status panel are read from the system metric MBeans. These same metric values can also be read directly from the MBeans using any JMX monitoring tool such as JConsole which is included with a Java Development Kit (JDK) or an enterprise monitoring solution such as IBM Tivoli Monitoring. The URL for the external JMX interface is,

```
service:jmx:rmi://serverName/jndi/rmi://serverName:9700/proxyserver
```

The external JMX interface is enabled by default and is configured to use port number 9700. You may also elect to secure the connection to the external JMX interface by configuring a user name and password that would be required before connections to the JMX interface are permitted.

You can manage both the TCP port number and the credentials for the external JMX interface using the Environment Properties within IBM Cognos Configuration.
1. Open IBM Cognos Configuration on an IBM Cognos BI server.
2. Within the Explorer Panel, click **Environment**.
3. Modify the External JMX Port and External JMX credential properties as required to fit your environment.

**Figure 8. IBM Cognos Configuration external JMX port and credential settings**

![IBM Cognos Configuration external JMX port and credential settings](image)

**Next Step**

Now that you are familiar with IBM Cognos BI system metrics, continue on to the document titled **System Management Methodology For IBM Cognos 10 - Setting Metric Thresholds** to understand how system metrics can be loaded into a reporting database and used for system trending reports.