IBM CORPORATION

WebSphere Business Monitor Checklist for V6.2

WebSphere Business Monitor Development Team

9/30/2010

This document presents checklists and tips to help you install, configure, and use WebSphere Business Monitor.

Table of Contents

Abstract

This document presents a list of information to help you install, configure, and use WebSphere Business Monitor.

Introduction

This document uses checklists to present information to consider when using WebSphere Business Monitor. The checklists are not exhaustive or ordered but provide tips to help you more fully exploit the capabilities of WebSphere Business Monitor.

A complement to this document is the Frequently Asked Questions (FAQ) about WebSphere Business Monitor, which is updated regularly and available online at the following URL:

http://www-01.ibm.com/support/docview.wss?uid=swg27017940

Installing

Before installing, the following checklists and tips can help you plan a successful WebSphere Business Monitor installation.

❑	Verify that your operating system and hardware (dedicated memory, hard disk space, and so on) match the requirements for WebSphere Business Monitor. See: http://www.ibm.com/support/docview.wss?uid=swg27013667
❑	For Linux or UNIX, before installing software, verify that the ulimit values are checked and set properly. See: http://www.ibm.com/support/docview.wss?t&uid=swg21381759
❑	When installing fix packs to a WebSphere Business Monitor environment that runs on a UNIX platform such as Linux or AIX, use a user ID that has read, write, and execute permissions to the file system being used for WebSphere Business Monitor.
❑	Consider installing a 64-bit version of WebSphere Business Monitor so that you can run with a larger JVM heap size. With additional memory, you can allocate more JVM heap space for use with WebSphere Business Monitor. The JVMs that benefit from more memory are: • The cluster that is running your monitor model applications. • The cluster that is running your CEI server. Note: With 64-bit JVMs, the JVM heap size can be larger than with 32-bit JVMs.

Monitor Server Runtime

This section contains helpful information for the Monitor Server runtime environments and covers both WebSphere Process Server and WebSphere Application Server environments.

❑	If you plan to use WebSphere Process Server in addition to WebSphere Business Monitor, decide on a recommended topology. There is an IBM Red Book "WebSphere Business Process Management V6.2 Production Topologies" (SG247732) that is a must read. See: http://www.redbooks.ibm.com/redpieces/abstracts/sg247732.html?Open

Tooling

This section contains helpful information for the Monitor tooling environment for both WebSphere Integration Developer and Rational Application Developer environments.

❑	Ensure that the machine has enough physical memory dedicated to WebSphere Business Monitor. It is a good idea to check and see what other applications might be running on this machine and consuming resources. Also make sure that your virtual memory settings have

	been set up properly. For one source of information, see: http://support.microsoft.com/kb/308417 In some cases, the WebSphere Business Monitor server will not start because of improper virtual memory settings.

Tips

- For best practices for using WebSphere Business Monitor, see the Redpaper "Best practices using WebSphere Business Modeler and Monitor" at the following URL: http://www.redbooks.ibm.com/abstracts/redp4159.html?Open
- To generate an EAR file for your monitor models, there are two valid approaches. The first is through the use of tooling and the second is through the use of headless deployment (the mmdeploy command). You can find information about both approaches in the information center.
- Do not export a monitor model EAR file from the WebSphere administrative console and attempt to deploy it on another machine. This is not supported and will not work.
- The monitor models that you develop within the tooling can be debugged using the debugger that comes with the tooling.

Configuring

Once the installation of WebSphere Business Monitor server has been completed, the following checklists can help with configuring the server and deploying monitor models.

❑	For most environments, use the queue-bypass method to improve performance and simplify the system configuration. If both queue-based and queue-bypass event distribution are configured, queue bypass is selected as the default event distribution method. However, if the Monitor database is geographically remote from CEI, consider using queue-based event distribution instead.
❑	If you are working in a secured environment with a remote CEI server and you are using queue-based CEI event distribution, you must add a user to the sender role of the corresponding foreign bus and destination every time a model or model version is deployed. For information about how to add the sender role, see http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/admin/model_install_secure.html
❑	Make sure the WebSphere Business Monitor scheduler function is correctly configured. After model deployment, review scheduler intervals for the model and update as needed.

	See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/config/default_dss_settings.html. Data Movement Service (DMS) and Cube Summary Tables (MQTs) must be enabled before they show up in the scheduled services. Both can be enabled after data has been collected, so there is no need to enable them immediately upon model deployment. DMS helps server and dashboard performance when there is a high event volume. Cube Summary Tables help the performance of the Dimensional and Report widgets when many instances are collected. In production environments, Alphablox Cube Caching should be turned on regardless of whether Cube Summary Tables are enabled. This will cache measures data in Alphablox across user sessions and improve dimensional query performance. At high event volumes, the large amount of data collected can begin to degrade performance. You can configure the scheduler to automatically purge and archive old completed instances. These instances could be imported into a duplicate WebSphere Business Monitor installation that is used for historical reporting and analysis. See the information center pages following the above link for more details on each of the scheduled services settings.
❑	Configure dynamic alerts. If you plan to use the Alert Manager to define dynamic alerts, it is not clear from the information center that additional configuration by the administrator is required. For example, if you intend to make use of alert notifications such as email alerts, the administrator must have configured SMTP. Configuration is also needed to take advantage of cell phone or pager alerts. For information about how to configure these options, refer to the following URLs: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/admin/actman_cfg_genl.html http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/admin/actman_cfg_vmm.html http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/admin/actman_cfg_ldap.html

Network Deployment Configuration

This section provides configuration information for a network deployment environment.

❑	When planning for an network deployment topology, always implement a fail-over capability.
❑	Test the connections to your database. In a network deployment topology, most databases

	are located on dedicated database servers. To verify connectivity: 1. Sign in to the WebSphere administrative console. 2. Click **Resources > JDBC > data sources.** 3. Set the scope to be at the cell level. 4. Select all data sources and click the **Test connection** button. 5. Validate that there are no connection errors. Remember the following considerations: • The user ID and the database permissions that the user ID has. • When you generate monitor model applications, database tables are created for each monitor model. This task is often left for the DBA to perform. • If you are using Oracle, pay careful attention to details regarding grants and getting permissions set correctly. This area frequently causes problems. For more information about the considerations around securing your database, refer to the following URL: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/security/sec_db_environ_ov.html
❑	Make sure that CEI is correctly deployed and configured. The information center states that "You must deploy and configure the common event infrastructure (CEI) to send and receive events." However, if you already have a WebSphere Process Server cell and intend to add Monitor to this cell, you already have a CEI Server installed and configured. You do not need to repeat this configuration task. To determine whether CEI has been configured: 1. Sign in to the WebSphere administrative console. 2. Click **Service Integration**. If **Common Event Infrastructure** is listed below it, then CEI has been deployed. 3. If you have run the **deployEventService** wsadmin task but have not restarted the deployment manager, **Common Event Infrastructure** will not appear on the WebSphere administrative console. (If in doubt, you can simply restart your deployment manager to be sure) Deploying and configuring CEI is a manual task that consists of running several wsadmin commands. These commands are documented in the information center. At a high level, the sequence of commands to run is: 1. deployEventService 2. configEventServiceDB2DB 3. enableEventService 4. setEventServiceJmsAuthAlias Deploying and configuring CEI also requires the creation of database tables for the CEI messaging engine and the Monitor messaging engine. Make certain that the CEI messaging engine database tables are created. If the JDBC resources or database tables are not created, your messaging engines will not start. The sequence of manual steps that must be followed can be found at the following URL: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.

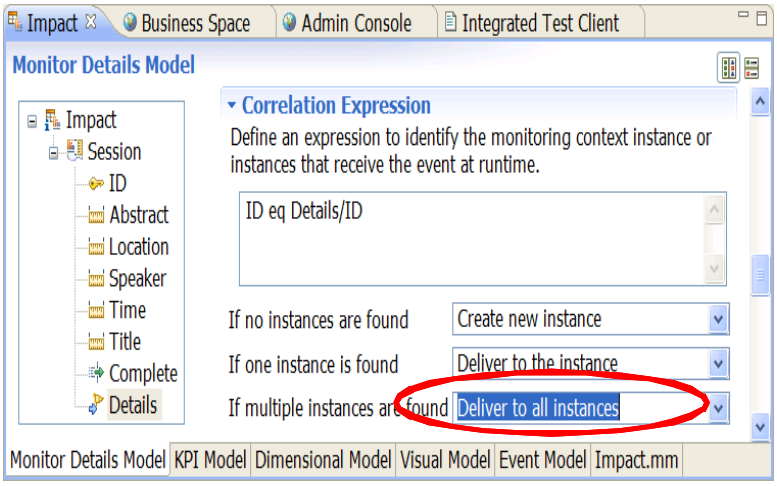
	monitor.install.doc/install/cei_deploy.html NOTE: At the present time, there is a missing step in the information center. After step 5 (Enable the CEI event service) and before step 6 (Restart your cell), run the **setEventServiceJmsAuthAlias** command, using the following parameters for a clustered environment: • $AdminTask setEventServiceJmsAuthAlias { -clusterName *monSupportCluster* –userName *monadmin* –password *monadmin* } • $AdminConfig save
❑	If your CEI bus fails to start, check the SystemOut.log file for errors such as: • SWSIS1524E: Data source, jdbc/com.ibm.ws.sib/monSuppCluster-CommonEventInfrastructure_Bus, not found. To resolve this error, create a JDBC data source and run the database scripts that create the CEI messaging engine data store. • CWSIT0073W: No intra-bus messaging engine authentication alias is configured. To resolve this error, run the **setEventServiceJmsAuthAlias** command as documented above.
❑	In a production environment, make sure that the event data store is disabled. When CEI is configured, the event data store is enabled by default, but this behavior is acceptable only for development and testing environments. In a production environment, it causes performance to suffer. To disable the event data store: 1. Sign in to the administrative console. 2. Click **Service Integration > Common Event Infrastructure > Event Service**. 3. Under Additional properties, click **Event Services**. 4. Click the link in the **Name** column to select the appropriate CEI event service. 5. Clear the **Enable event data store** check box. 6. Click **OK** and then save your changes to the master configuration.
❑	Make sure that you configure the Monitor bus and messaging engine data store. If your SystemOut.log file contains a message similar to "CWMLC0260E: Monitor SIBus [MONITOR.mymon620Cell01.Bus] not defined", the Monitor bus has not been configured. There are two ways to configure the bus and messaging engine. • Run a script: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/install/bus_create.html • Use the WebSphere Business Monitor configuration wizard available on the WebSphere administrative console to configure the Monitor bus and messaging engine as well as other components: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/admin/config_msgengine.html
❑	Make sure that your Monitor bus and messaging engine start successfully.

	When the Monitor messaging engine has successfully started, the SystemOut.log file contains a message similar to "CWSID0016:: Messaging engine monMECluster.000-MONITOR.wbi602dCell01.Bus is in state Started." If your Monitor bus doesn't start successfully, verify that the Monitor messaging engine database tables have been created. If you see a message similar to "CWS1S1532: The table, MONITOR.SIBOWNER, is missing.", the tables have definitely not been created. Note: In the above sample messages, "MONITOR" was the schema name and your schema name may vary.
❑	In a cross-cell environment, remember: • If security is enabled in one cell, it must be enabled in the other cell. • If security is enabled, you must enable server-to-server trust (SSL) and share LTPA keys. To enable SSL, see the following URL: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/admin/cfg_ssl_montowps_cei.html To share LTPA keys, see the following URL: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/security/ltpa_cfg.html
❑	In a network deployment environment, remember that the startup and shutdown order of your servers is important. To avoid the locked event situation or the locked or in-doubt state, always stop the monitoring environment using the following procedure: 1. Stop the cluster hosting the monitor model applications. 2. Stop the cluster hosting the CEI server. 3. Stop the cluster hosting the messaging engine. When starting your environment, start the clusters in the following order: 1. Start the cluster hosting the messaging engine. 2. Start the cluster hosting the CEI server. 3. Start the cluster hosting the monitor model applications.

Authoring Monitor Models

This section discusses considerations to take into account when developing a monitor model. The first checklist contains some best practices to keep in mind when authoring monitor models.

❑	To update a monitor model after it has been deployed once, update the model version time stamp in the Monitor Model editor but do not change the model ID. When you have finished making changes, use the Monitor Model editor to generate a new EAR file. See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.

	monitor.admin.doc/admin/model_version.html
❑	When creating a monitor model, do not use a time-based trigger to terminate a monitoring context. To see why this practice is not recommended, consider this scenario: • Normal event processing for a given process completes in two hours. • You define a termination trigger to terminate an instance after eight hours. • There is an unexpected outage and WebSphere Business Monitor does not consume events for two days. Unpredictable and certainly unexpected results will occur.
❑	When using recurring wait-time triggers in a monitor model, keep the following considerations in mind: • The length of time to wait. For example, does a trigger really need to fire every two minutes? • The number of instances that will be present in your environment. If there are thousands, tens of thousands, or perhaps millions of instances present, consider the processing time associated with each trigger firing. • The number of recurring wait-time triggers. Numerous triggers compound the effect of not considering the wait-time length and the number of instances.
❑	Be careful when selecting **Deliver to all instances** as the Event delivery option for multiple correlation matches. If you have many active instances and the event could correlate to a high percentage of them, performance can be affected.  For example, if you have 1 million active instances, an event that correlates to 1/3 of them, arriving once every 5 minutes, results in over 1,000 event deliveries per second. This rate could potentially tax the server to the point that it could not keep up with its other inbound events.
❑	Decide whether to use event groups or monitoring contexts when generating a monitor model.

A monitoring context is a container for WebSphere Business Monitor artifacts such as metrics, stopwatches, triggers, and events. By default, when you create a monitor model, you must have at least one monitoring context. An event group is optional and is simply a folder inside a monitoring context. An event group can only contain inbound events and is used to logically organize a set of related events inside a monitoring context. It is purely a visual construct and is not represented in the monitor model XML file.

When you generate a monitor model using the Generate Monitor Model wizard, you can create a monitoring context to hold your events or put them in an event group that belongs to a parent monitoring context. The more monitoring contexts you select, the more deeply nested your monitor model becomes. The more event groups you select, the flatter your monitor model structure.

A cube is automatically generated for every monitoring context. The more cubes in your model, the slower your system performance. Therefore, reduce the number of monitoring contexts to get better performance, and design your monitor model to minimize the number of cubes created. **Best Practice:** Delete cubes if there are no dimensions or measures associated with them. Alternatively, eliminate any child monitoring contexts that are not needed.

When deciding whether to use a monitoring context or event group, in general you can let the logical structure of the process and the business requirements drive the decision. For example, if the process contains a loop or a subprocess, decide which construct to use. When you are generating a monitor model for a large process or processes, the following hints may help you decide which to choose.

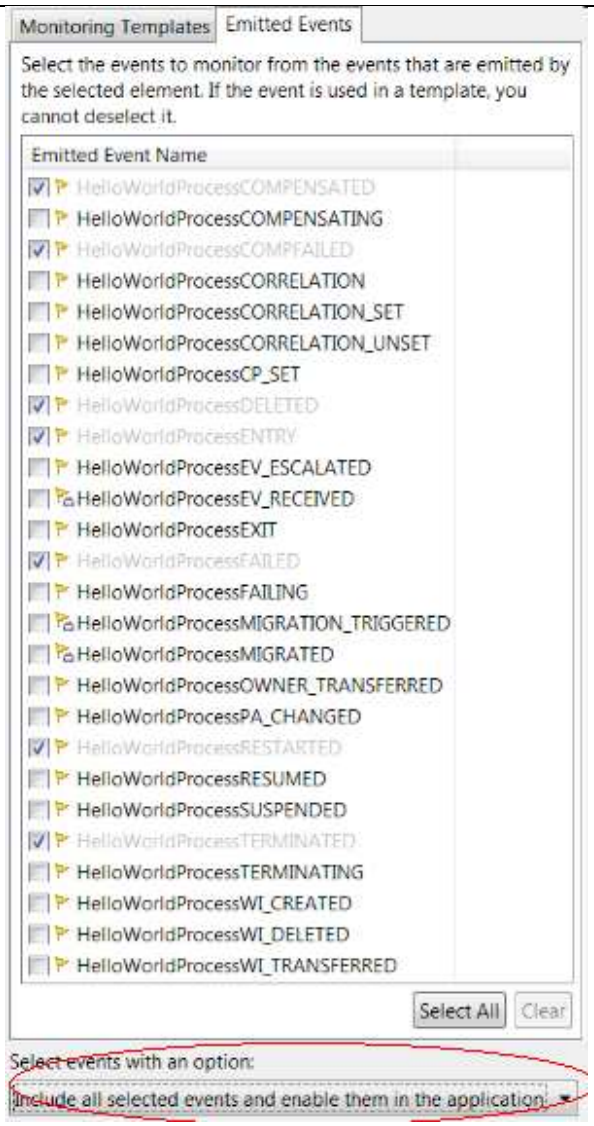
When to use event groups:

- A flatter model performs better than a more deeply nested model. A flatter structure creates fewer cubes. Flatter is better from a performance perspective, for both event processing and Data Movement Services (DMS) because it is faster to manage one long database row than multiple shorter database rows.
- If the customer does not want to drill down in the Instances view in the dashboard, use a flatter monitor model with all the metrics in the same monitoring context. Consider using event groups instead of nested monitoring contexts if the users prefer to see all the instance data in one table in the Instances widget or if they prefer KPIs that filter on a combination of metrics that are in parent and child monitoring contexts. Also consider using event groups if users might want to perform dimensional analysis on a combination of metrics in parent and child monitoring contexts.
- If there is a sub-process that happens only 0 or 1 times, you can use either an event group or a monitoring context, but there are typically more advantages with a flatter structure.

When to use monitoring contexts:

- If you need to monitor multiple processes in the same monitor model and you need to aggregate the instance data across the processes and correlation data is available, consider using a root-level monitoring context to contain the process monitoring contexts.
- If there is a nested sub-process that loops, you will need a child monitoring context

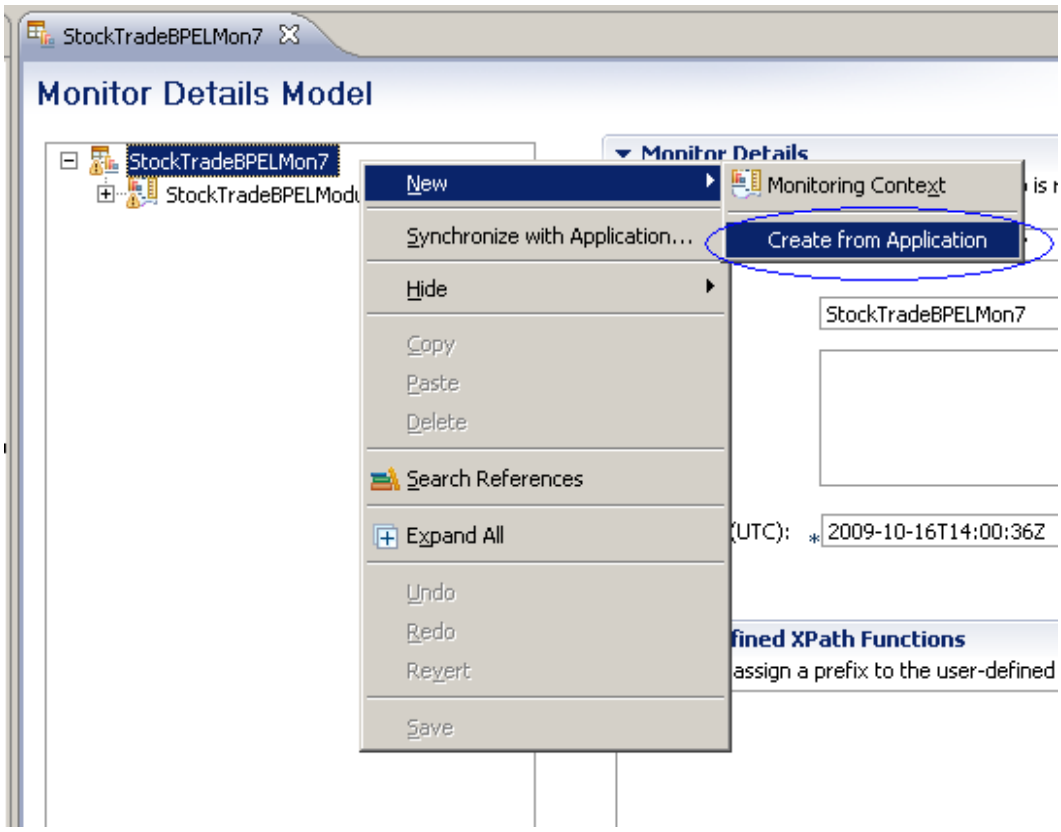
	to handle the instance data for each loop iteration. • Each monitoring context is represented as a table in the WebSphere Business Monitor database. There is a 32KB limit per row in the database. (This is a restriction in the supported databases for row size.) If you use a flatter monitor model with a single monitoring context, you risk hitting this limit. The more metrics, counters, stopwatches, and triggers added to a monitoring context, the larger the row will be in the database. Be aware that each metric, counter, stopwatch, and trigger represents a column in the monitoring context table in the database. You will not typically hit this limit, but it is possible, especially if you are storing long string data in your metrics. If you hit the limit, you could put the longest string metrics into a nested child monitoring context.
❑	Decide on naming conventions for your metrics. Be aware when you are naming the metrics that the metric name is what you will see on the dashboards. Keep metric names as short as you can. Each monitoring context metric is represented as a column in the monitoring context database table and the column name is a truncated form of the metric name.
❑	Decide on naming conventions for your applications. Windows path-length restrictions can be exceeded when long application names are used or when the WebSphere Application Server is installed in a deep directory structure. Be careful to keep both the model name and the installation path as short as possible to avoid possible problems.
❑	Limit the events that you enable when authoring a monitor model. Turn on event emission only where truly necessary for business activity monitoring. WebSphere Business Monitor is for business monitoring and should not be used for IT monitoring. For example, enabling emission for all possible event types might cause 100 events to be emitted per process instance even though your monitor model has KPIs that care about only 10 of them. Turning off unneeded events improves the performance of both the emitting application and the monitor model. Best Practice: Turn on only the events that are used in the selected templates and variables. The following screenshot shows how to use the Generate Monitor Model wizard to select only the events you need and then how to enable only those events to be emitted at runtime:



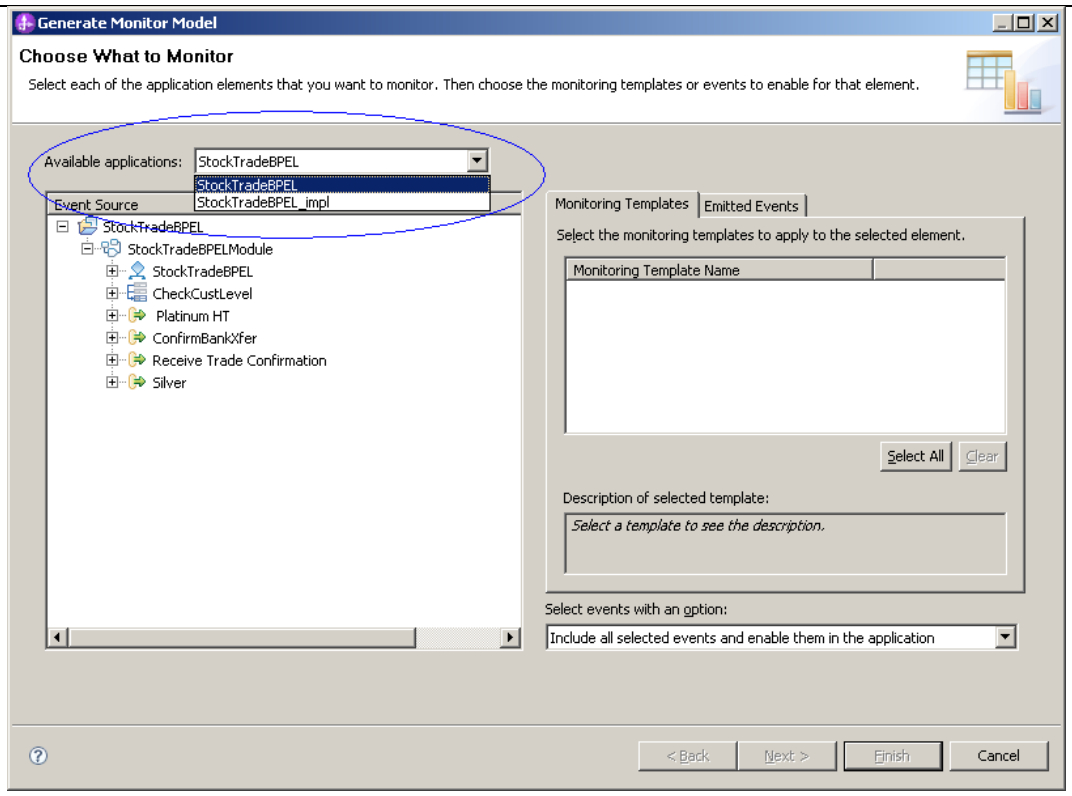
An alternative approach is to first identify the data that is needed in the dashboard and then deduce which events are required to be able to monitor that data. Often, dashboard data is based on the content of the business object payload. Since the payload can sometimes be very large, define smaller variables throughout your WebSphere Integration Developer application to hold the specific payload data you are interested in monitoring at different points in the process. Then, insert a Java snippet into key locations of your process to copy the data from the business object payload into your smaller variable. Finally, emit a BPEL variableChanged event based on this smaller variable which contains only the necessary payload fields. Following this guideline, you can greatly reduce unnecessary event traffic and limit it to exactly what is required in the dashboard.

❑	If you need to monitor multiple processes applications that reside in different WebSphere Integration Developer projects in a single monitor model, consider using the following procedure.

1. Use the Generate Monitor Model wizard to generate your monitor model for the processes you are interested in monitoring.
2. Open the resulting monitor model in the Business Monitoring perspective. Right-click the monitor model and select **New > Create from Application**.

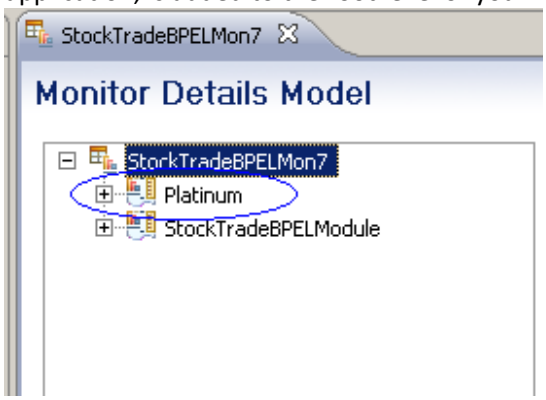


3. Open the resulting monitor model in the Business Monitoring perspective. Right-click the monitor model and select **New > Create from Application**.
4. In the Choose What to Monitor window, click the drop-down list to view the available applications for monitoring. All the related WebSphere Integration Developer projects will be listed.



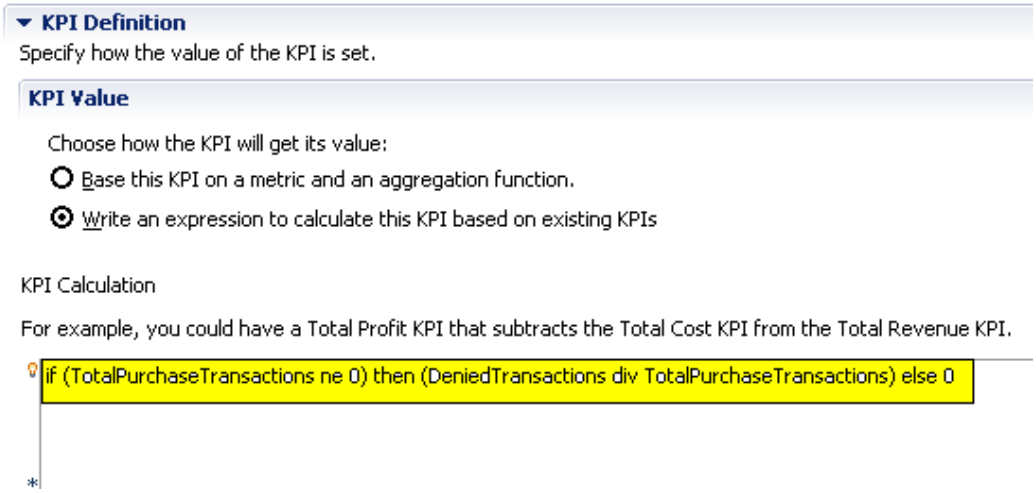
5. Select the application to add to your existing monitor model. The Generate Monitor Model wizard displays the monitoring options for the new application.
6. Step through the wizard, selecting the events and templates to include in your monitor model, and click **Finish**.

A new monitoring context, which includes the options you selected for monitoring this application, is added to the root level of your monitor model.



❑ When defining calculated KPI expressions, ensure that the KPI values in a denominator calculation are not zero. **Best Practice**: Test the value of the denominator as part of the calculation expression and set the KPI value accordingly. This practice will avoid divide-by-zero errors in KPI calculations.

For example, in the following KPI calculation, the highlighted expression is used to test the

	value of TotalPurchaseTransactions and make sure it is not 0 before computing the value of the KPI. 
❑	Decide whether to use a low level and a high level monitor model, or a single monitor model. Reasons to use a low-level and a high-level model: • **Isolate Dashboard Data:** The high-level model can be used to track monitor specific data while the low-level model is used to track BPEL events. Because this causes a more complex relationship where the model developer has to manage the events between the models, consider using WebSphere Business Modeler to generate the low-level and high-level models for you. In this case, when you export the Modeler project to WebSphere Integration Developer, be sure to select the option to Export to 2 monitor models. • **Decrease Model Complexity for Business Users:** Use this approach when you want to isolate and track only the Monitor-specific data for the dashboards and shield business users from the complexity of the BPEL model. If a single model is going to be too big (a lot of metrics, for example), it could become too complex for business users to be able to efficiently choose their metrics in their custom dashboards. • **Simplify Change Management:** If the BPEL process is changing frequently but those changes do not affect the metrics in the dashboard, using two monitor models enables you to make changes to the low-level model without impacting the high-level model that the dashboards are based on. • **Monitor Multiple Processes:** If you are monitoring multiple processes or applications from other WebSphere Integration Developer projects in a single monitor model, you need to use a high-level monitor model to synchronize and aggregate the instance data across those processes. For example, if your generated monitor model contains sibling process monitoring contexts at the root monitor model level, you need a high-level monitor model to aggregate instance data across these sibling monitoring contexts. • **Monitor Different Event Sources:** When you are monitoring different event sources, it is generally best to split up monitor models based upon the event source. Each model is autonomous, a separate application with its own input event

stream.

Reasons to use a single model:

- **Ease of Use:** If you are just beginning to develop a monitor model for a set of processes, it is easier to put everything into a single monitor model. However, as you become more proficient in monitor model development, you might see the need for aggregating the metric data in a secondary high-level model for the reasons stated above.
- **Maintenance:** A single model means less maintenance. Things are simplified with a single model (less moving parts for versioning, for example). If changes to a BPEL process require changes to the metrics in the dashboard, the high level model must also be updated, which could mean double maintenance.
- **Performance:** Fewer monitor models yield better performance.
- **Decreased Event Traffic:** If you anticipate a high-event traffic rate from the BPEL process, using the high-level and low-level approach will generate even more event traffic because you need to manage events between the models. A single model is better if you want to control, limit, or decrease event traffic.
- **Efficiency:** If you can specifically isolate the BPEL events you are interested in monitoring and they contain the metric data you need for your dashboard, you can develop a single custom monitor model. Be aware that when you generate a monitor model using the Generate Monitor Model wizard, if you choose to use only the events you have enabled, the default templates might not be available for selection any more.

Note: There is a 32KB size limit in the database for each monitoring context. It is possible that a single monitoring context could become too large. If that does occur, you will see an error in the Monitor Model editor.

Alternatives:
An alternative to using low-level and high-level models is to simply use the global process monitor model that can be automatically installed during profile creation. This model monitors BPEL processes running on WebSphere Process Server. It detects and tracks any BPEL process that sends events to WebSphere Business Monitor. Processes are discovered dynamically, and monitored both at the definition level and at the execution level. The generated dashboards are more for process administrators than for business users. For more information, see :
http://publib.boulder.ibm.com/bpcsamp/monitoring/globalprocessmonitor.html

For a technical deep dive into the global process monitor model, see:
http://www.ibm.com/developerworks/websphere/library/techarticles/1002_frank/1002_frank.html

A second alternative is to generate the low-level and high-level models (using WebSphere Business Modeler for example), and then edit the generated low-level model to strip it down to a single model which only monitors the very specific events and metrics you are interested in.

References:

	Connecting low-level and high-level models: http://publib.boulder.ibm.com/infocenter/dmndhelp/v7r0mx/topic/com.ibm.wbimonitor.help.toolkit.doc/mme/connectingmodels.html Multi-module monitoring with the WebSphere BPM Suite: http://www.ibm.com/developerworks/websphere/library/techarticles/1003_lakshmanan/1003_lakshmanan.html
❑	When refactoring in the toolkit environment, consider the following tips: • If you are working on WebSphere Integration Developer artifacts, you must be in the Business Integration perspective. • If you are working on WebSphere Business Monitor artifacts, you must be in the Business Monitor perspective. • If the artifact is already open in the corresponding perspective, it is a good idea to close the artifact and then re-open it.

Performance Considerations

How many monitor models can be deployed on a server?

It is very hard to give a precise answer to this question since there are numerous factors that need to be considered. Examples of factors that need to be considered are:

- Event rates –
  - How large are the business objects?
  - How many events are typically sent for a single instance?
  - What is the volume of events?
    - Do events arrive in a batch (for example, overnight during off business hours) or do events arrive spread out over time?
  - Are there peak points in time where volumes are high (for example, certain times of day or certain months of the year)?
- Monitor model complexity – Are there large numbers of metrics involved? Are numerous timers involved?
- The hardware used and how the topology is set up – Be sure to read the IBM Red Book "WebSphere Business Process Management V6.2 Production Topologies" (SG247732) mentioned earlier. See: http://www.redbooks.ibm.com/redpieces/abstracts/sg247732.html?Open

There are some general practices that can help with performance. Observe the following:

- In order to be as accurate as possible in assessing your performance, the system being measured should be nearly identical to the system that will be used in production. This way, there are no surprises later on.

- The biggest improvements come from simply tuning the database. Database tuning changes tend to yield significant improvements and should be made before attempting to change any of the per-model settings referenced in the next item.
- What does the live memory in the Java heap under load look like? There are some per-model settings that can be set in the administrative console. Information about these settings can be found in the information center. See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/admin/model_cfg_runtime.html
- What does the CPU consumption look like under load? If the CPU bottlenecks, response times can get large quickly.
- Server startup time. If there are many monitor models per server, there may be additional delays getting the server started or stopped .

So what can be done to achieve good performance? Review the following items in the check list (which is in no particular order) for guidance.

❑	Tune the database. Database tuning can make a huge difference in performance. After material changes to the database tables, perform RUNSTATS to collect new table statistics that will help the query optimizer. Increase transaction logs to handle large instance or KPI history imports and exports, and large DMS executions (such as when DMS enablement has been deferred until after there are millions of instances in the source MCT tables). Increase the size of buffer pools to improve dashboard query performance. For small to medium WebSphere Business Monitor installations, it may be possible to have a large percentage of data in buffer pool memory, thereby avoiding costly table scans on disk. The monitor model create schema DDL includes a best-guess set of indexes based on sortable columns, dimensions, and KPI filters. If there are a large number of monitor instances, it is recommended that a DBA analyze the tables during normal user queries to determine if additional indexes may help.
❑	Use the right performance or tuning techniques for WebSphere Business Monitor. A Redpaper "WebSphere Business Process Management 6.2.0 Performance Tuning" is available at the following URL: http://www.redbooks.ibm.com/abstracts/redp4551.html?Open
❑	In production environments, have a DBA create the WebSphere Business Monitor database during product installation, and execute the monitor model create schema DDL scripts after monitor model deployment. This way the privileges of the WebSphere Business Monitor runtime user can be kept to a minimum. For securing the database, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/security/sec_db_environ_ov.html

	Note that the GRANT privilege scripts provided in the information center link above should be executed immediately after the installation or model deployment DDLs are executed, so that the runtime user ID has correct privileges from the start.
❑	For using Alphablox, reduce the number of years pre-configured for dimensional analysis to the range of dates that you need. This will improve memory utilization and server performance, since each day in the range gets cached in each cube as a dimension member. See : http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/admin/ablox_dim_table.html
❑	Investigate more Alphablox tuning tips at the following URL: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/config/abx_tune.html Note that WebSphere Business Monitor no longer uses the Alphablox cube schedule as documented here, but instead uses the Monitor scheduled services cube caching service. (This information center error was corrected in later releases.) See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/data/abx_cubecaching_manage.html
❑	Address distributing the deployment of monitor models so that performance is acceptable. One possibility is to define multiple clusters and deploy models to different clusters. Another possibility is to use a custom high availability policy. To specify a custom high availability policy for the monitor model applications, the first step is to review the WebSphere Application Server information center topic at the following URL: http://publib.boulder.ibm.com/infocenter/wasinfo/v6r1/topic/com.ibm.websphere.nd.multiplatform.doc/info/ae/ae/trun_ha_newpolicy.html The second step is to consider how to define your policy. The match criteria for the policy uses the following name/value pairs. You can be as specific (specify a specific version) or as general (specify just a component) as you need. **Name** / **Value** / **Notes** type / WBM_Monitor / Identifies WebSphere Business Monitor product component / MODERATOR / WBM component model / model id / version / model version / Matches String.valueOf(model version) **WARNING**: You **must** specify a One-of-N policy to ensure that only one member is active at a time.
❑	Tune Alphablox. Information about tuning Alphablox for WebSphere Business Monitor can

	be found at the following URL: http://www.ibm.com/support/docview.wss?uid=swg27013234
❑	Limit the number of Alphablox cubes in the monitor model. Keep the lowest number that meets the customer requirements. If there are many cubes, disable either the cross-version or the version-specific cubes. Notice that there can be 0 to 1 cubes for each monitoring context. If they are not needed, delete the cubes at that level. Cubes are memory hogs in Alphablox. For information about disabling and deleting cubes, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.install.doc/config/abx_tune_cube_disable.html
❑	Consider using multi-threaded processing mode. Events are processed more efficiently in multi-threaded mode. In addition, this mode supports the error queue (failed event queue) and can support event reordering (which you need when using asynchronous event emission). To use multi-threaded mode: • All inbound events must contain the key to the root monitoring context. • All correlation expressions (including for events for children and grandchildren monitoring contexts) must refer to that key to the root monitoring context. For information about processing strategies in WebSphere Business Monitor, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/admin/strategies_process.html
❑	For performance reasons, do not monitor the process variables. Whenever the variable is read for a process instance, regardless of whether it is updated; an event is emitted, which can create excessive event traffic. Instead, monitor only the events with the business object payload. In WebSphere Integration Developer, select the events that contain the payload data needed for the dashboard metrics. Make sure the event content is set to Full to be able to access the payload data. If the Business Object is very large or contains confidential information, consider two possible options: 1. Insert a Java snippet into key locations of your process to copy the data from the business object payload into your smaller variable. 2. Add an ASSIGN activity to your process and copy only the fields that you are interested in monitoring into another variable. Then, emit a BPEL variableChanged event based on this smaller variable which contains only the necessary payload fields and is only referenced once.
❑	For better performance when using dashboards, consider the following tips: • Queries should be indexed on the database. • Assign at least 1 GB of memory to the Monitor database to keep all the indexes in

	memory. Increase the size of buffer pools to improve dashboard query performance. • On the instance widget, large number of rows will impact browser performance. Take into account that the further you page the worse the performance will be, so consider filtering and sorting instead of deep paging. • KPIs can be cached at the KPI level starting in 6.2. It is possible to alter caching and evaluation behavior based on latency needs and individual KPI performance: o 'Wide' KPIs that aggregate a large percentage of instances (for example, total sales amount for all instances this year) should be cached. The user probably doesn't need to know annual sales every 5 minutes. If KPI caching can be set to at least 60 minutes, enable KPI history and let the scheduler perform the KPI calculations in the background. o 'Narrow' KPIs that aggregate a targeted set of instances (for example, total sales amount for instances in the past hour) don't need to be cached because the user probably wants to see frequent updates within the hour. This type of KPI is typically serviced by efficient database indexes. To establish the KPI cache refresh interval per KPI, go to the KPI Manager. Click Actions > Properties and go to the Other tab. Click the check box and set an interval in minutes to override the default KPI cache interval (in minutes) that is configured on the runtime configuration of the monitor model. The KPI cache refresh interval is located in the KPI tab on the runtime configuration of a monitor model. See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/admin/model_cfg_runtime.html • To increase the response time for the Dimensions and Report widgets, consider caching the cube measure values. http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/data/abx_cubecaching_manage.html
❑	Archive and purge old instances in the database on a one-time or scheduled basis to increase dashboard efficiency and performance. Database query performance degrades as the number of rows (instances) increases. For information about unscheduled archiving and purging, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/data/purgearchive_ov.html If you want to see the archived data in your Monitor widgets, you can install another Monitor server to point to your archived data. You can enable additional widgets manually for this archived data in your Monitor server that contains the current data. To enable Business Space widgets for multiple end points, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.websphere.wbpm.bspace.config.620.doc/doc/tcfg_bsp_multiple_endpoints.html
❑	Enable data movement services (DMS) to duplicate the database tables.

	Because each monitoring context represents a table in the Monitor database, under high volumes there can be contention between the Monitor Server operations and the dashboard for the use of these tables. For better performance, enable DMS, which will split the monitoring context table in two, one table for real-time event processing and the other table for dashboard reporting, To enable DMS, see: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.admin.doc/data/dms_manage.html
❑	In a production environment, make sure that the event data store is disabled. When CEI is configured, the event data store is enabled by default, but this behavior is acceptable only for development and testing environments. In a production environment, it causes performance to suffer. To disable the event data store: 1. Sign in to the administrative console. 2. Click **Service Integration > Common Event Infrastructure > Event Service**. 3. Under Additional properties, click **Event Services**. 4. Click the link in the **Name** column to select the appropriate CEI event service. 5. Clear the **Enable event data store** check box. 6. Click **OK** and then save your changes to the master configuration.
❑	Delete the "All Events" Event Group Name to improve performance. Follow these steps: • Sign in to the administrative console • Click Service Integration >Common Event Infrastructure > Event Service. • Under Additional properties, click Event Services. • Click the link in the Name column to select the appropriate CEI event service. • Under Additional properties, click Event groups. • Check Select for the All events Event Group Name • Click Delete and then click Save to save your configuration change.

Troubleshooting

❑	There is a Frequently Asked Questions (FAQ) about WebSphere Business Monitor document on the IBM Support portal that is updated monthly and contains numerous troubleshooting hints and tips. See: http://www.ibm.com/support/docview.wss?uid=swg27017940
❑	To prove that the topology is correctly configured and to be sure that a monitor model is deployed and will consume events, use the new diagnostic aid available in WebSphere Business Monitor v6.2.0.2 Fix Pack. See: http://www-

	01.ibm.com/support/docview.wss?rs=802&context=SSSRR3&dc=DB550&dc=D420&dc=DB530&dc=D410&dc=DB510&dc=D430&q1=diagnostic&uid=swg24023969&loc=en_US&cs=utf-8&lang=en
❑	To debug your monitor model, use the Integrated Test Environment to send events and validate that metrics are set properly and that your model is operating correctly. See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.dev.doc/testing/integrated_test_client.html
❑	To troubleshoot issues with your monitor model, use the debugger that comes with WebSphere Business Monitor. See: http://publib.boulder.ibm.com/infocenter/dmndhelp/v6r2mx/topic/com.ibm.btools.help.monitor.dev.doc/testing/startingdebugger.html
❑	If you are deploying a monitor model to a remote CEI server and get an error message such as “CWMAX4203E: Error occurred getting the list of Event group profile names” and that message also includes a message like “ADMC0017E: Could not create an RMI connector to connect to host myHostName at port 2810”, the most likely cause is a configuration problem. Examine the deployment manager SystemOut.log and SystemErr.log files. If there is an SSLHandShakeError exception, you must exchange the SSL certificates between the two cells.
❑	To verify that a monitor model deployment is successful, perform the following steps: 1. In the WebSphere administrative console, click **Applications-> Monitor Models** and make sure that the status in the **Deployment** column is "OK." If the status is "Error", a problem has occurred during deployment that will need to be investigated. 2. From the same panel, click the link in the **Version** column for the model and check the following things: 1. The "Dashboards enabled" and "Schema created" steps have green check marks next to them. 2. If the model has Alphablox cubes, then "Alphablox cubes created (optional)" has a green check mark next to it. 3. The "CEI distribution mode" box has a value that starts with the word "Active". 3. The "Dashboards enabled" step is related to updating "repository" database tables in the Monitor database schema created at product installation time (not the model-specific schema for a particular model). Some of these database tables can be manually inspected for verification purposes; for example, the following tables: • META_MONITOR_CONTEXT_T • META_MONITOR_METRIC_T For example, the META_MONITOR_METRIC_T table lists metrics associated with the model version. For the MODEL_ID entries that match the model, look at the VERSION column that has a date and time stamp for the model version (not the rows with zero). Then review the METRIC_ID column to see a list of metrics for the model.

	4. The "Schema created" step is related to creating the database schema for the model version that is being installed. If the schema creation script is run automatically at model deployment time, you can export the schema creation script for review to validate the database artifacts created. To export the script, click **Applications > Monitor Models**, click the link in the **Version** column, click **Manage Schema**, and click **Export Create Schema Script**. You can then check any of the DDL statements in the file against the artifacts created. 5. The "Alphablox cubes create" step is optional since a model can operate without Alphablox cubes being created. However, you can list the cubes in the Alphablox Administration Console to check that the number of cubes created matches the number in the model. The Alphablox Administration Console can be accessed at http://<host>:<WC_defaulthost port>/AlphabloxAdmin. From the **Administration** tab, click **Cubes** to list the cubes. 6. If there are multiple possible event emitters (each with its own CEI), make sure the monitor model was deployed to point to the correct emitting CEI location. To check the location, click **Applications > Monitor Models**, click the link in the **Version** column, click **Change CEI distribution mode**, and click **Change CEI configuration**. The scope of the "Event group profile list name" indicates the emitting CEI corresponding to the model. 7. In the emitting CEI location, check that an event group for the monitor model has been created. Click **Service integration > Common Event Infrastructure > Event service > Event services > *Name of the event service* > Event groups**. There should be an event group that is named similarly to the monitor model. 8. If you are deploying a second version of a monitor model, pay special attention to the "CEI distribution mode" for the model. The second version should have a distribution mode that starts with "Active". The first version of the model (the older one) should have a distribution mode that starts with "Active" and then includes "no new MC instances". 9. Make sure it is possible to successfully start the model from the administrative console.
❑	Before uninstalling a model**, verify that the emitting CEI server is running.** If the CEI server is not running, the enterprise application associated with the model will be uninstalled but monitor model artifacts will remain in the database. If artifacts remain in the database, then either the entire model (all versions) must be purged or a specific model version must be purged.
❑	If you find an issue with a second or third model version deployment in a production environment, perform the following steps: 1. Record information about the sequence of steps followed before and after the issue so that as much context and state information as possible is available. While traces were probably not enabled at the time of the issue, collect as much of the "must gather" information as possible. The document that lists "must gather" information for 6.2 is available at the following location: http://www-01.ibm.com/support/docview.wss?uid=swg21405891. The files listed in point three under

	"General Diagnostic Information" are needed. Collect the contents of the database tables mentioned in point 3a under "Manually collect component-specific MustGather data". 2. Decide whether to open a PMR. If the deployment attempt is keeping the model version that was previously running in production from operating, it is a good idea to open a PMR with the initial state information about the failure. Consider an example with model versions n and n+1. If version n+1 fails to deploy, checking whether newly emitted events are still being consumed by version n is important. A situation where neither version n nor version n+1 is consuming newly emitted events is more urgent than other types of situations. Inspect the CEI distribution mode for model version n and see if it is "Active" and does not include "no new MC instances"; if the status is "Active" without "no new MC instances" at the end, newly emitted events are still consumed by version n. 3. Once the cause of the issue is determined (or if there is a strong theory), prepare for trying to deploy the model version again. Remove Alphablox cubes for the model version, remove the database schema for the model version, and uninstall the enterprise application for the model version. Then run a script to remove artifacts ("purge") a single model version. See the following technote for an example script: http://www-01.ibm.com/support/docview.wss?uid=swg21406844 The script in the technote above calls Monitor Lifecycle Services MBean commands. See the following location for information about the Lifecycle Services MBean commands: http://publib.boulder.ibm.com/bpcsamp/v6r1/monitoring/jythonScripts/javadoc/public/com/ibm/wbimonitor/lifecycle/spi/mbeans/LifecycleServicesMBean.html. Running the script in the technote is the best option because it does validation. However, individual Lifecycle Services MBean commands could also be run similar to the following. • Establish a connection to the Lifecycle Services MBean. wsadmin>set ls [$AdminControl completeObjectName type=LifecycleServices,*] • Run the following Lifecycle Services MBean command to disable dashboards. wsadmin>$AdminControl invoke $ls disableDashboards { "<model ID>" <model version string - in YYYYMMDDHHMMSS form> } • Check the version listing for the model to make sure that the green check mark for the "Dashboards enabled" step is gone. • Also check the META_MODEL_STEP_T database table to make sure that the IS_COMPLETE column is set to '0' for the "enableDashboards" step. The "disableDashboards" step should also have IS_COMPLETE set to '1'. • Run the Lifecycle Services MBean "confirmUninstall" method. wsadmin>$AdminControl invoke $ls confirmUninstall { "<model ID>" <model version string

	- in YYYYMMDDHHMMSS form> } • Use the Lifecycle Services MBean to unregister the model so it no longer shows up in the administrative console. wsadmin>$AdminControl invoke $ls unregister { "<model ID>" <model version string - in YYYYMMDDHHMMSS form> }