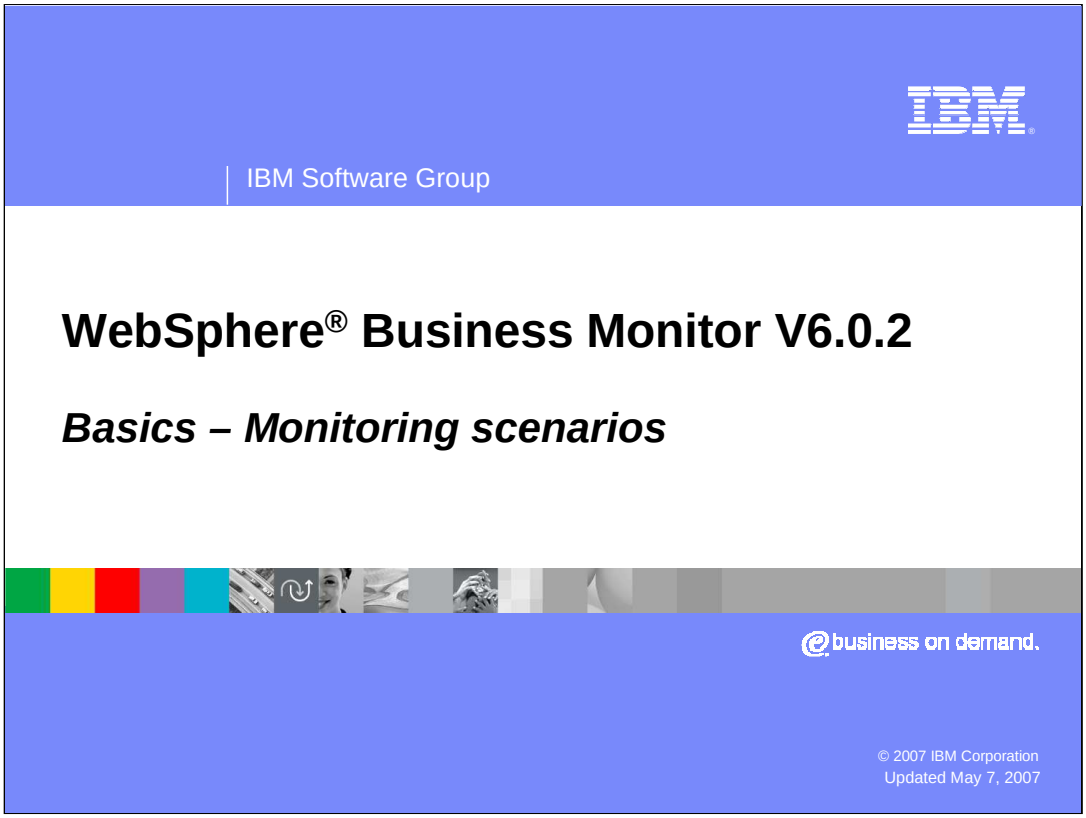
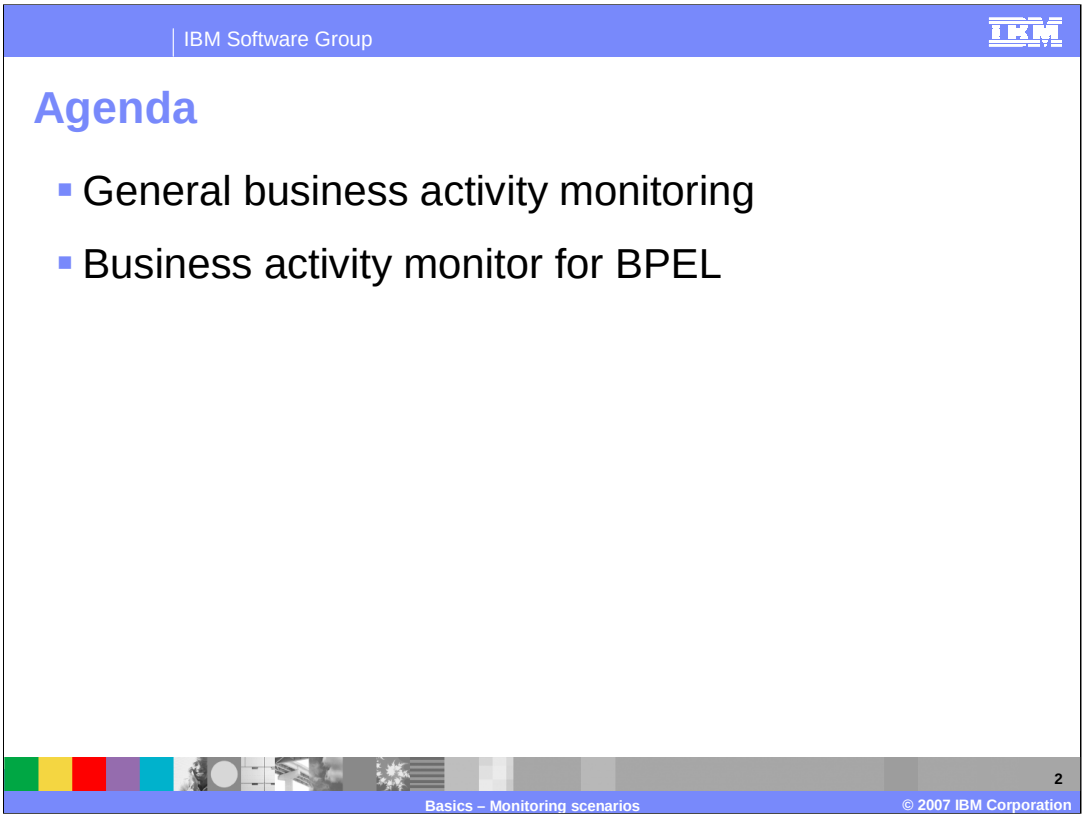
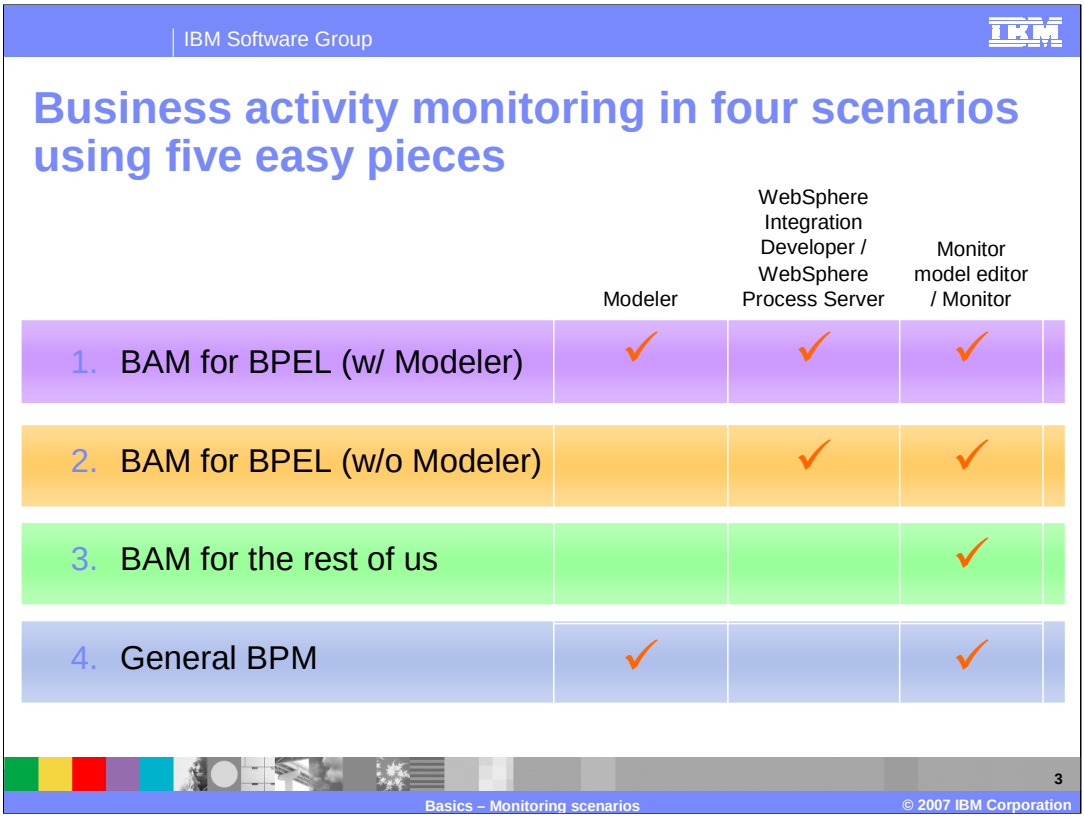


This presentation should give you a good understanding of the process for developing Monitor models for different scenarios.

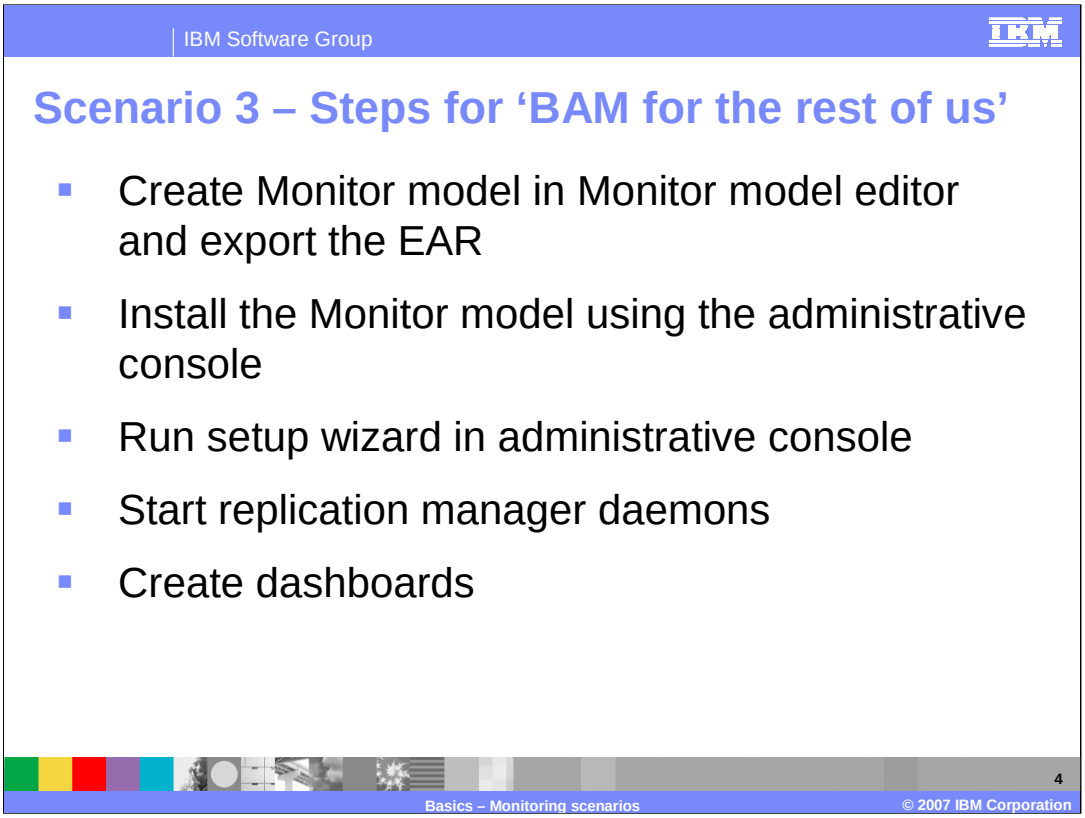


This is the agenda for the presentation.

Different scenarios will be discussed for business activity monitoring, including general event based monitoring and monitoring business processes using BPEL.

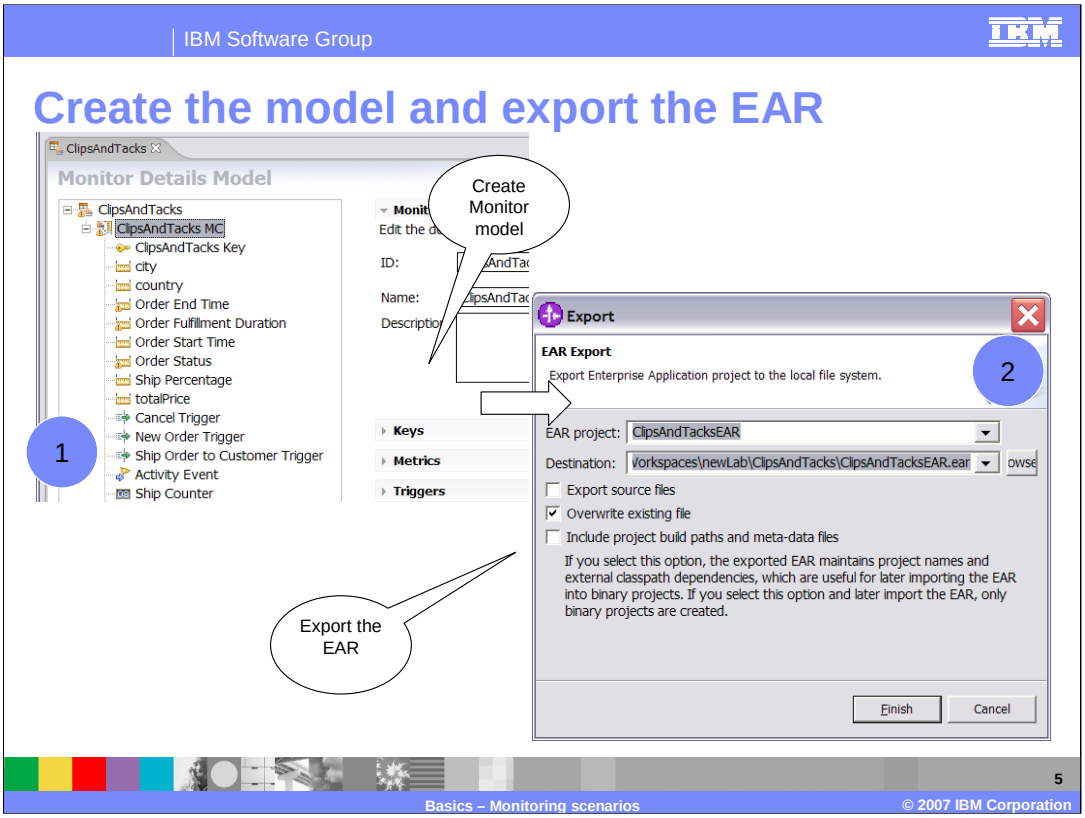


This chart shows the four scenarios that are supported in version 6.0.2, namely monitoring BPEL with or without the use of Modeler, and monitoring non-BPEL sources with or without the use of Modeler.



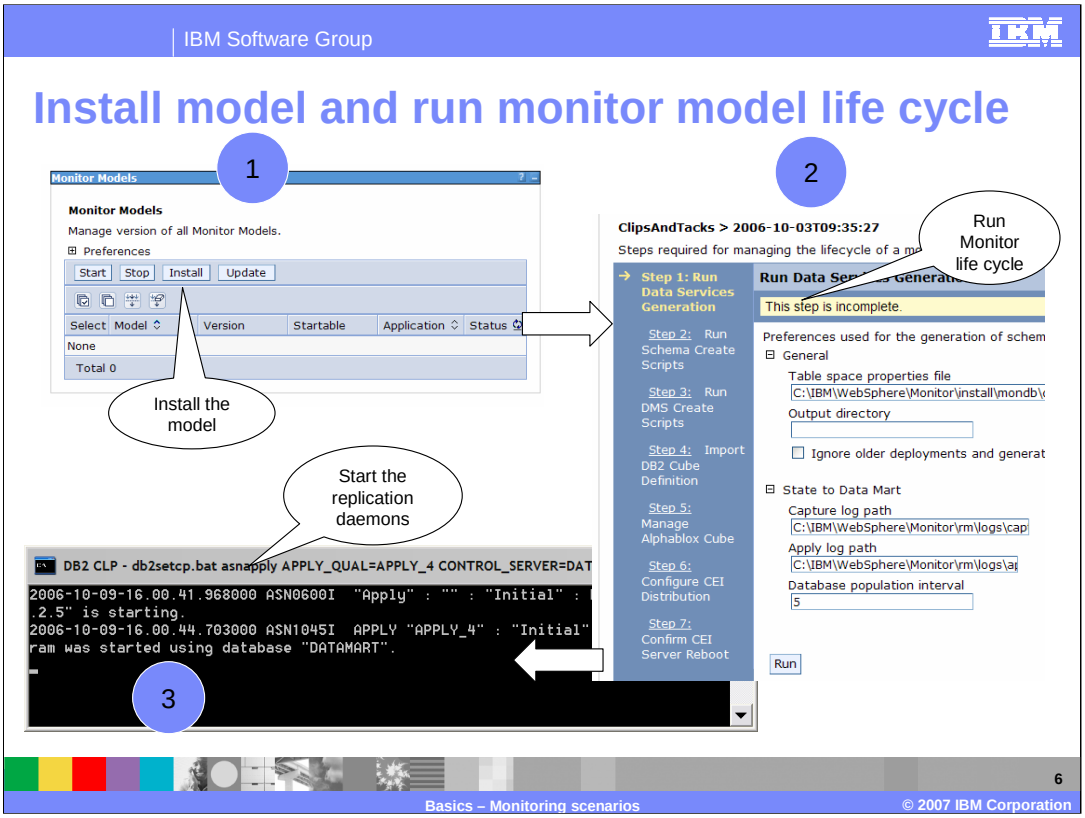
For Scenario 3, these are the steps to create and deploy a monitor model.

You create the model using the Monitor Model Editor, then export the EAR. Then you install the Monitor model using the administrative console of the Monitor Server, and you also run the setup wizard there to complete the steps of the model deployment life cycle. When that is complete, then you start the replication daemons and finally create dashboards on the Dashboard server.



The next set of slides show some screen prints of the steps to create and deploy a monitor model.

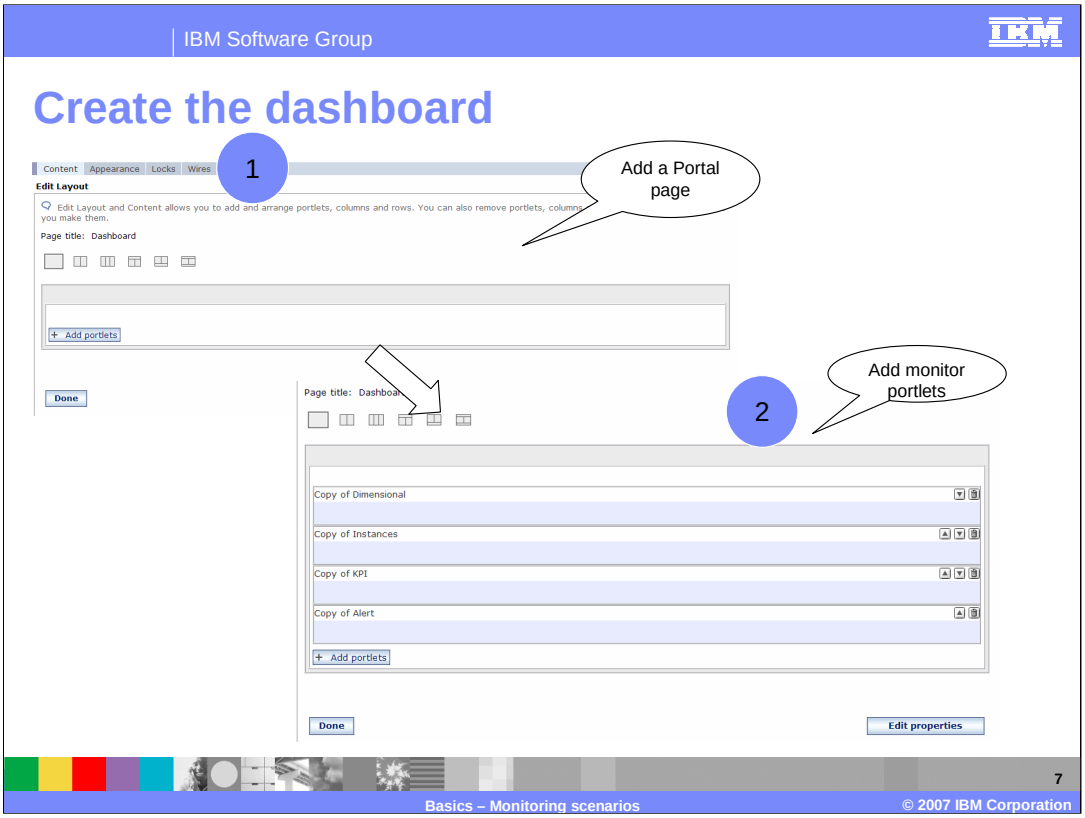
First, you build the model using the Monitor Model Editor in WebSphere Integration Developer. Then you switch to the J2EE Perspective, and export the EAR.



Using the administrative console of Process Server, you install the EAR.

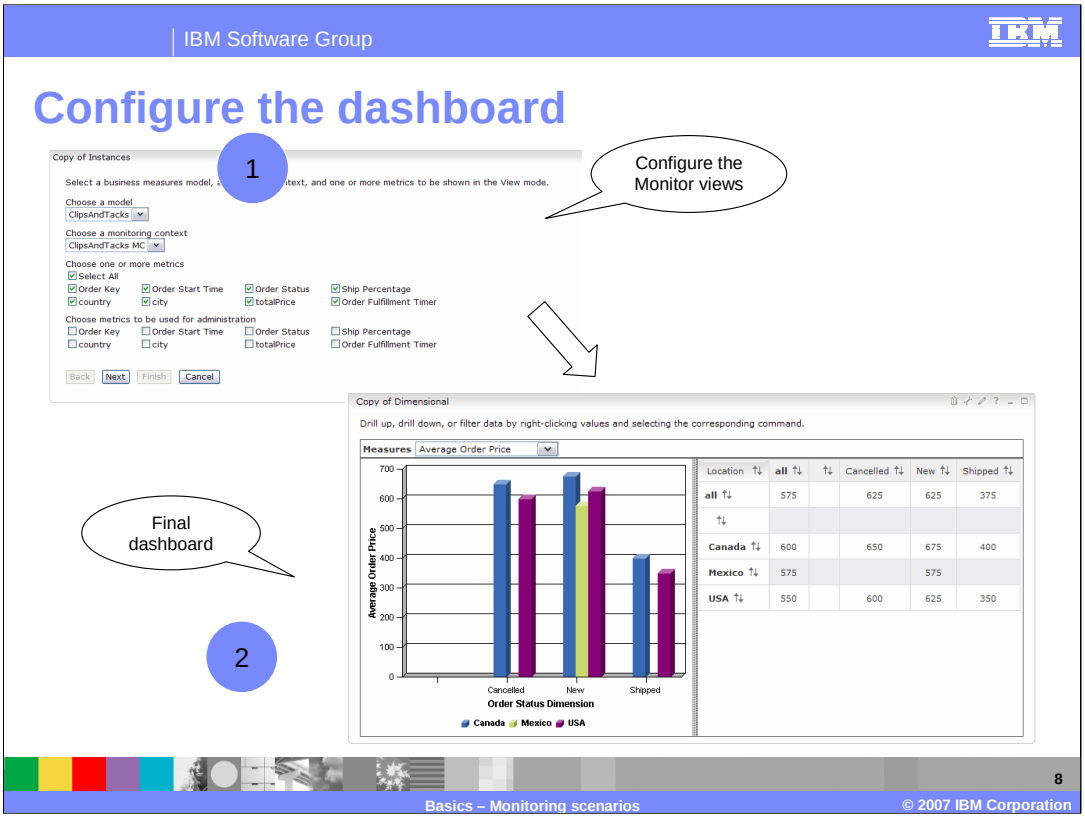
Then you run through the seven steps of the Monitor life cycle.

Next you start the daemons for replication. There will be several windows that are opened, one for each capture and apply server.



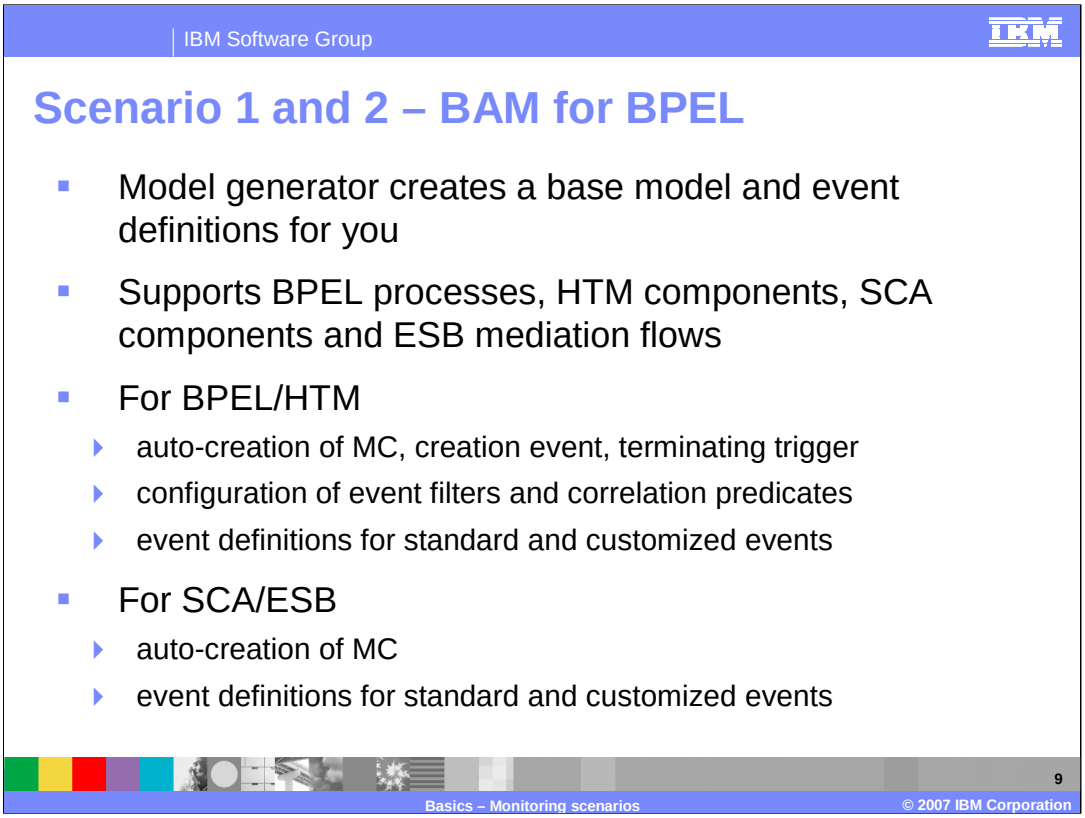
To create the dashboard you add a portal page to contain your portlets.

Then you add Monitor portlets to the Page to display whichever views are appropriate for your dashboard.



For each view that you added to the dashboard page, you will need to configure it to identify which models to display and specify other view specific information.

After the configuration of the views then you can view monitored information in your new dashboard.



The BPEL monitor model generator auto-creates a basic monitoring model that is capable of monitoring a particular BPEL process **at an instance level**. This means that you will have a single monitoring context instance for each process instance that is created at runtime. It would be difficult for you to create the model yourself because the filtering and correlation is based on system-defined information in the event definitions, and not business payload information.

For BPEL and HTM components, the generator creates event definitions and a base model that contains MC creation constructs, MC termination constructs, and correlation information.

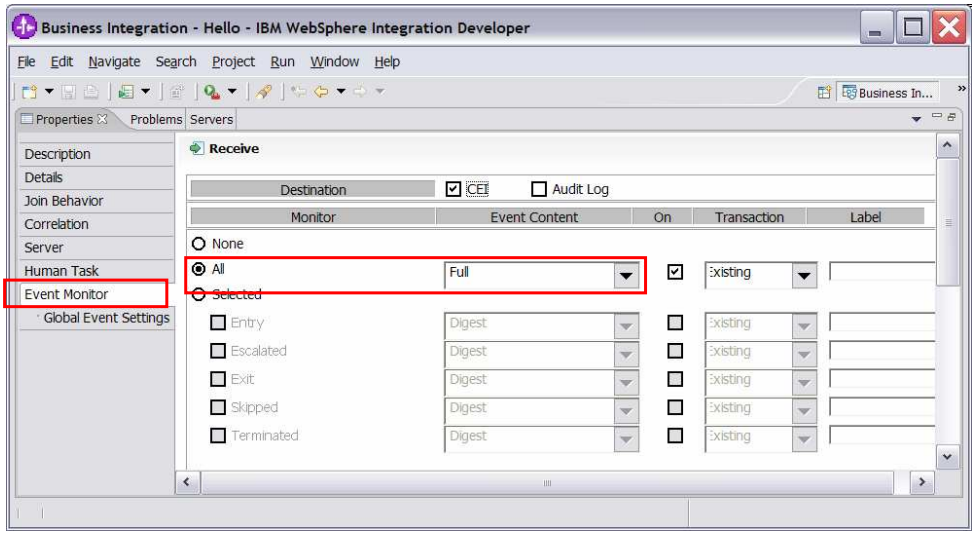
For SCA and ESB components, the generator creates the event definitions and a base model that contains only a monitoring context, but no creation, termination nor correlation information. You will have to create this yourself.

Scenario 1 and 2 – Steps for 'BAM for BPEL'

- Scenario 1 only: create process/monitor model in Modeler and export
- Implement the process model in WebSphere Integration Developer
- Change Event Monitor setting to 'Full' for BPEL events that you want to monitor
- Auto generate Monitor event definitions
- Auto generate base Monitor model
- Add metrics, KPIs, dimensions to your model
- Build, export, deploy the Monitor EAR
- Build, export, deploy the BPEL EAR

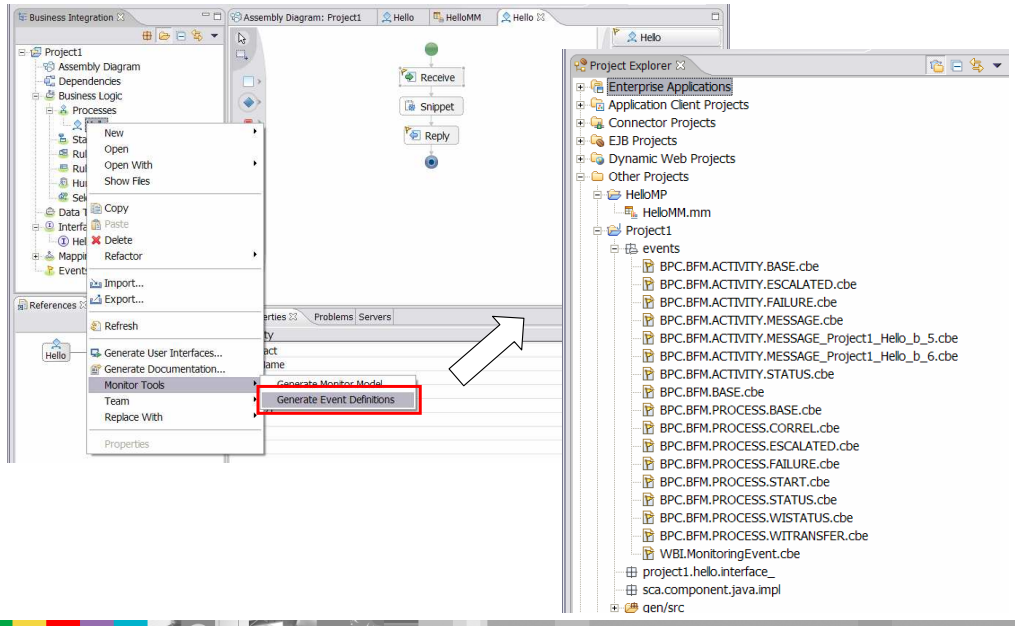
These are the steps to use to create a deploy BPEL monitor models. For scenario 1 only, you would use Modeler to create the process model and the monitor model. Then you implement the process model in WebSphere Integration Developer. You should review the Event Monitor tab for the BPEL elements to decide which events to create for Monitor usage. Then you use the monitor generator functions in WebSphere Integration Developer to auto-generate the event definitions and the base monitor model. You will need to add the KPIs and business measures to this model, then deploy the Monitor model EAR and the BPEL process EAR.

Event monitor setting

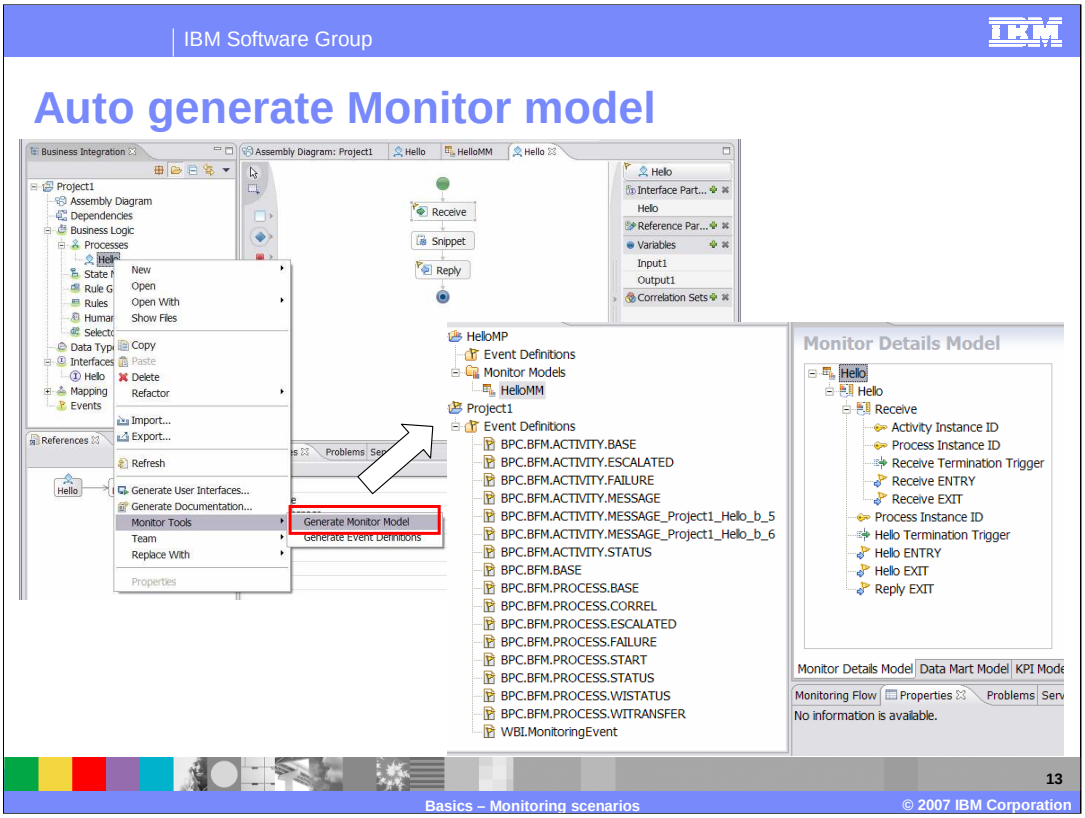


This is a screen capture of the Event Monitor tab in WebSphere Integration Developer. You may select All to send all events for the select BPEL element, or you can choose option Selected so then you can pick which events to send to Monitor. To ensure that you send the payload, change the event content setting from Digest to Full.

Auto generate Monitor event definitions

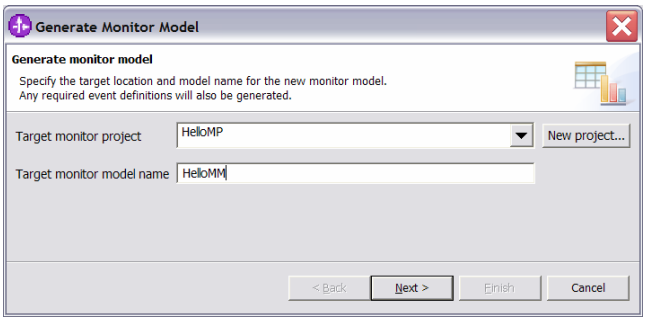


To use the Monitor event generator, you right click on the BPEL process in the Business Integration view, then pick Monitor Tools then Generate Event Definitions. On the right side of this slide you see that the event definitions have been added to the project explorer view.



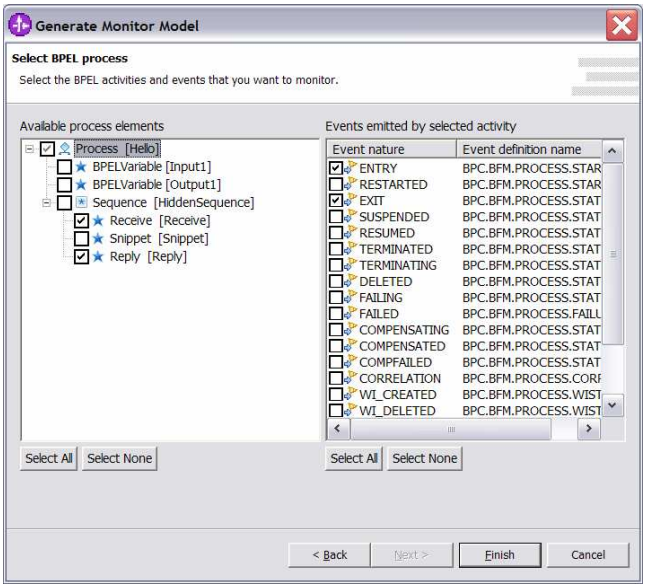
To use the Monitor model generator, you right click on the BPEL process in the Business Integration view, then pick Monitor Tools then Generate Monitor Model. On the right side of this slide you see that a base monitor model has been created for you in the Business Monitoring perspective. And in that model you can see that there is a monitoring context, with key, inbound events and termination triggers.

Enter model project and name



When you select to generate a monitor model for BPEL, this is the dialog that prompts you for the monitor project and monitor model name.

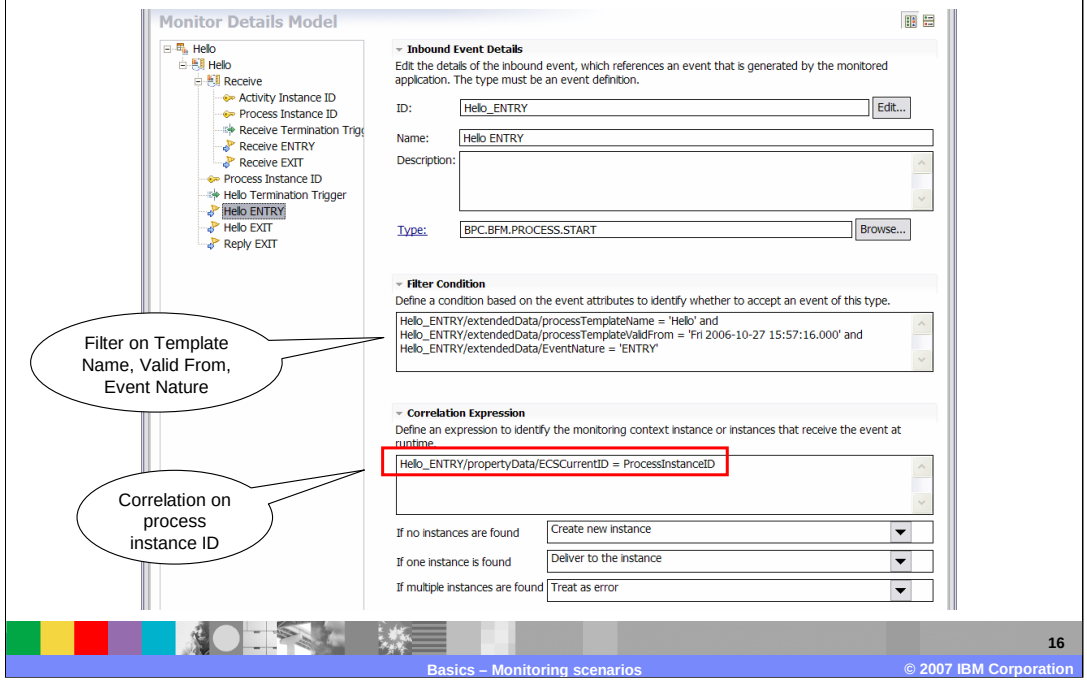
Select BPEL elements and events



After you enter the monitor project and model name then the next page allows you to customize the monitor model that is generated by selecting the particular BPEL constructs to monitor. In addition, you can select the individual events to be monitored for each construct.

The tree on the left side of the dialog shows a hierarchical representation of the BPEL process. Selecting an item in the BPEL process tree displays the event information for each event that the selected construct is capable of emitting at runtime. Only events that can be emitted are shown on the right, which means that any events that are not selected in the Event Monitor tab are not shown.

Process filters and correlation

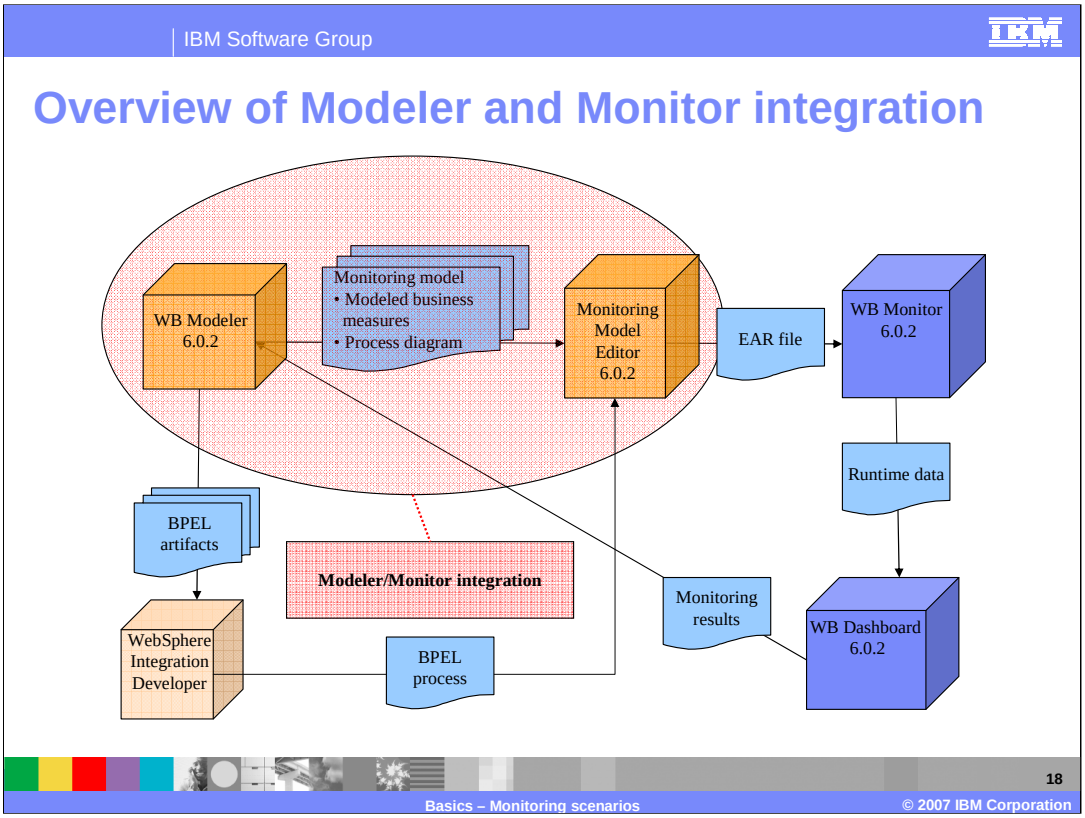


This is a Monitor details view in the Monitor Model Editor after auto-generating the monitor model for BPEL. For the inbound event for the process start, the filter condition has been set to access only certain events. The events are filtered by process template name, valid from date and event nature. Also, the correlation expression has been created so that the process instance ID is used to correlate events to monitored instances.

BPEL monitoring contexts

- Default Model contains MCs for each activity
- If data is not specifically needed at the activity level, move needed events up to the process level
- Advantages
  - Simpler, fewer contexts to maintain
  - By keeping metrics in one context, simplifies cubes, dimensions, and KPIs
  - Depending on changes to BPEL model, may not require any changes to monitoring model
- Disadvantages
  - More initial work to change and update events

When you use the monitor model generator, you will get a monitoring context for each activity. If you would like to simplify the model, you could remove the child monitoring contexts and move the data for the child MCs into the parent monitoring context. This would make for a simpler model and a simpler cube setup, but the disadvantage is certainly additional upfront work to setup the model.



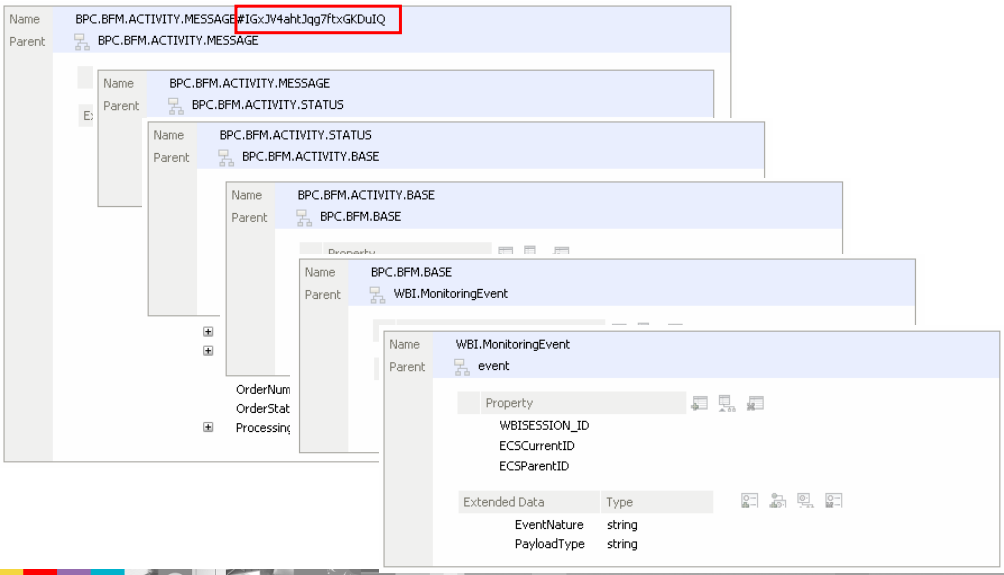
This slide depicts the integration of WebSphere Business Modeler and WebSphere Business Monitor. In Modeler you create the BPEL for the business process, and a high level Monitor model containing your KPIs using the BMV. In WebSphere Integration Developer you import the BPEL from Modeler then auto-generate a monitor model. The problem here is that there are two input arrows to the Monitor model editor. The KPI model comes from the BMV and another model comes from WebSphere Integration Developer for the auto-generated monitor model.

Model management – BMV and BPEL

- Modeler BVM Model contains KPIs
- Generated BPEL model contains MCs, Events, Key Metrics
- Two options
  - Merge the models
    - Advantage - One Model to maintain, deploy
    - Disadvantage - Merge KPI model every time a new BPEL Model is generated
  - Use both models and use events from Monitor model editor model to feed the KPIs in the BMV Model
    - Advantage - Generation of new BPEL Model has no impact on KPI Model
    - Disadvantage – Additional events required to pass data

Since you may have two sources for your monitor models, one from the BMV containing your KPIs and another which was auto-generated from the BPEL, then you need to decide how to manage this situation. One option is to merge the two models into a single consolidated model. The advantage to this approach is that you will only have one model to maintain and deploy into Monitor server. The disadvantage is that you will have to re-merge the models anytime the BPEL changes and you try to re-generate a new model from the BPEL. Another option is to keep both models intact, but to use inbound and outbound events to feed the data from the BPEL model to the BMV model so that the KPIs in the BMV model can be calculated. The advantage is that this means that you could generate a new monitor model from the BPEL and it would limit the impacts to the BMV model. However, there will be a performance impact due to the extra event processing.

Sample BPEL event definition – Activity message



When using the Monitor tools in WebSphere Integration Developer, you can automatically generate the event definitions that you would like to use in the Monitor model. This slide shows the hierarchy of event types that are used for the activity message event. There are six levels in the hierarchy but note that the last level inherits from event which is the main common base event parent definition.

Also note the mangled name on the initial activity message, which means that this event definition contains payload. If you do not select to have payload delivered, then this event definition would not be generated.

Sample BPEL event – Activity message

Standard Properties

Context Data

Extended Data

Name	Value
version	1.0.1
globalInstanceId	CEFD808060D780EB0CA1DB6F402B857C00
extensionName	BPC.BFM.ACTIVITY.MESSAGE#IGxJV4ahtJqg7ftxGKDulQ
localInstanceId	
creationTime	2006-11-08T15:45:33.632Z
severity	10
msg	
priority	
sequenceNumber	14
repeatCount	
elapsedTime	
contextDataElement / WBISESSION_ID / contextValue	192.168.88.128:ClipsAndTacks:processes/orderhandlingfuture1/Orde
contextDataElement / ECSCurrentID / contextValue	_AI:9004010e.c83ec7c3.7fa7573f.1aa6004a
contextDataElement / ECSParentID / contextValue	_PI:9003010e.c83ec4e5.7fa7573f.1aa6003a
extendedDataElement / PayloadType	full
extendedDataElement / EventNature	ENTRY
extendedDataElement / BPCEventCode	21007
extendedDataElement / processTemplateName	OrderHandlingFuture1
extendedDataElement / processTemplateValidFrom	Thu 2005-12-01 17:00:00.000
extendedDataElement / activityKind	21 - KIND_INVOKE
extendedDataElement / state	3 - STATE_RUNNING
extendedDataElement / bpelId	1009
extendedDataElement / activityTemplateName	ReviewOrder_InputCriterion
extendedDataElement / activityTemplateId	_AT:9002010e.c82884ba.7fa7573f.1aa60021
extendedDataElement / message	
extendedDataElement / message / Customer	
extendedDataElement / message / Customer / CustomerNumber	12345
extendedDataElement / message / Customer / CompanyName	ABC Finance Ltd.
extendedDataElement / message / Customer / ContactFirstName	Marc
extendedDataElement / message / Customer / ContactLastName	Shankaran

Here you see a sample activity message Common Base Event. There are three sections in the definition. The standard properties are common to every common base event and contain information such as the extensionName which identifies the event type, and other information such as event creation time. The context data section provides correlation information, and ECSCurrentID will store the process instance ID. The extended data section contains the payload for the event, so this is where you would extract business payload for calculation of metric and KPI values in your monitor model.

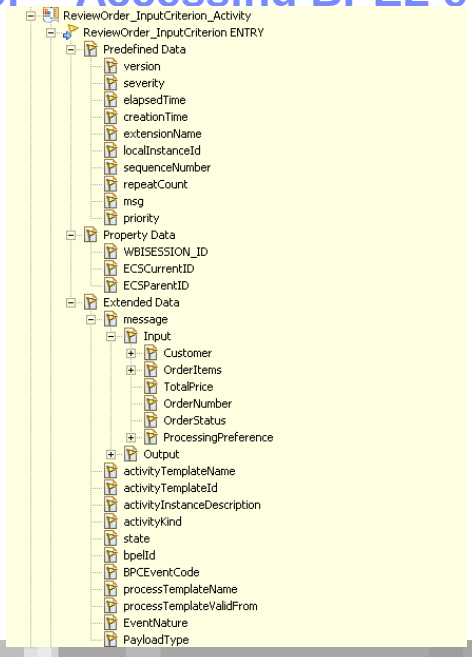
This is a screen capture of the Common Base Event browser which is part of the Process Server administrative console.

Monitor model editor – Accessing BPEL event data

Standard Properties

Context Data

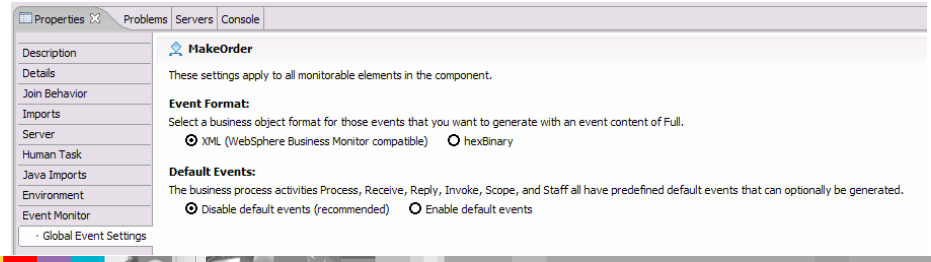
Extended Data



Here you see a BPEL event definition as it looks in the Monitor Model Editor. The three sections of data are shown. The standard properties are identified as 'Predefined Data', the context data is shown as 'Property Data' and the extended data is identified as 'Extended Data'.

Business object shredding

- A capability to allow BOs to be converted into a set of extended data elements
- Settings are in Event Monitor tab
  - Full – all payload is converted
  - Digest – a string description of the BO is given
  - Empty – no payload at all
- If you update this setting, regenerate the event definitions and monitor model
  - If you do not, MCs for the activities may not be created at runtime because process generates BPC.BFM.ACTIVITY.MESSAGE#<mangledName> but monitor model references BPC.BFM.ACTIVITY.MESSAGE, or vice versa
- Turn on shredding



Business objects are automatically converted into a set of extended data elements when placing the payload onto a common base event, and this is known as shredding. The settings that define how this done is identified in the Event Monitor tab in WebSphere Integration Developer. For each event, you can specify full, digest or empty. Full means that the entire payload is converted. Digest means that just a description of the business object is provided. Empty means that the payload is not placed on the event.

If you update the event monitor settings, you need to regenerate the event definitions and the monitor model. This is due to the fact that the event definitions that include payload will have some name changes which will affect the monitor model that you already created. For example, the activity message name includes a mangled name if it is carrying payload.

Make sure that you click the XML event format option in the Global Event Settings to ensure that the payload is properly formatted for Monitor.

BPEL events- How BPEL is translated to .mm

<process>	Monitor root element + Event Model, DataMart Model, KPI Model, Visual Model, Monitor Details Model + Monitoring context (top-level)
<scope>	Monitoring context
<while>	Monitoring context
<invoke>	Monitoring context
<receive>	Monitoring context
<pick>	Monitoring context
<wait>	Monitoring context
inline human task (<staff>)	Monitoring context
<sequence>	Inbound events on the nearest parent monitoring context
<flow>	Inbound events on the nearest parent monitoring context
<switch>	Inbound events on the nearest parent monitoring context
<reply>	Inbound events on the nearest parent monitoring context
<throw>	Inbound events on the nearest parent monitoring context
<empty>	Inbound events on the nearest parent monitoring context
<invoke> (Java™ Snippet)	Inbound events on the nearest parent monitoring context
<assign>	Inbound events on the nearest parent monitoring context
<rethrow>	Inbound events on the nearest parent monitoring context
<compensate>	Monitoring context
<terminate>	Inbound events on the nearest parent monitoring context
<variable>	Inbound events on the nearest parent monitoring context

This slide shows how the various BPEL elements relate to Monitor Model elements. The main BPEL process becomes the parent monitoring context in the monitor model. The activity elements such as scope, invoke, receive and pick become child monitoring contexts underneath the process monitoring context in the monitor model. The other BPEL elements become inbound events in the monitoring model and are correlated to the nearest parent monitoring context.

Events that have payload

- BPEL events
  - Invoke
    - ENTRY, EXIT
  - Staff (inline)
    - EXIT, OUTPUTSET FAULTSET
  - Receive, Reply, Pick
    - EXIT
  - Variable
    - CHANGED
- HTM events (component)
  - OUTPUTSET, FAULTSET
- SCA events
  - ENTRY, EXIT
- WebSphere Enterprise Service Bus events
  - CUSTOM

Some BPEL event definitions have payload and some do not. This slide shows you the event types that include payload. If you are designing your monitor model and you need to extract business data for calculating your KPIs, then you can use these event types for that reason.

BPEL creation and termination events

The following table shows the event natures that are available for each monitorable BPEL activity. Constructs with either **green** or **red** squares map to Monitoring Contexts in the Monitor Model. **Green** squares indicate context creation events. **Red** squares indicate context termination events.

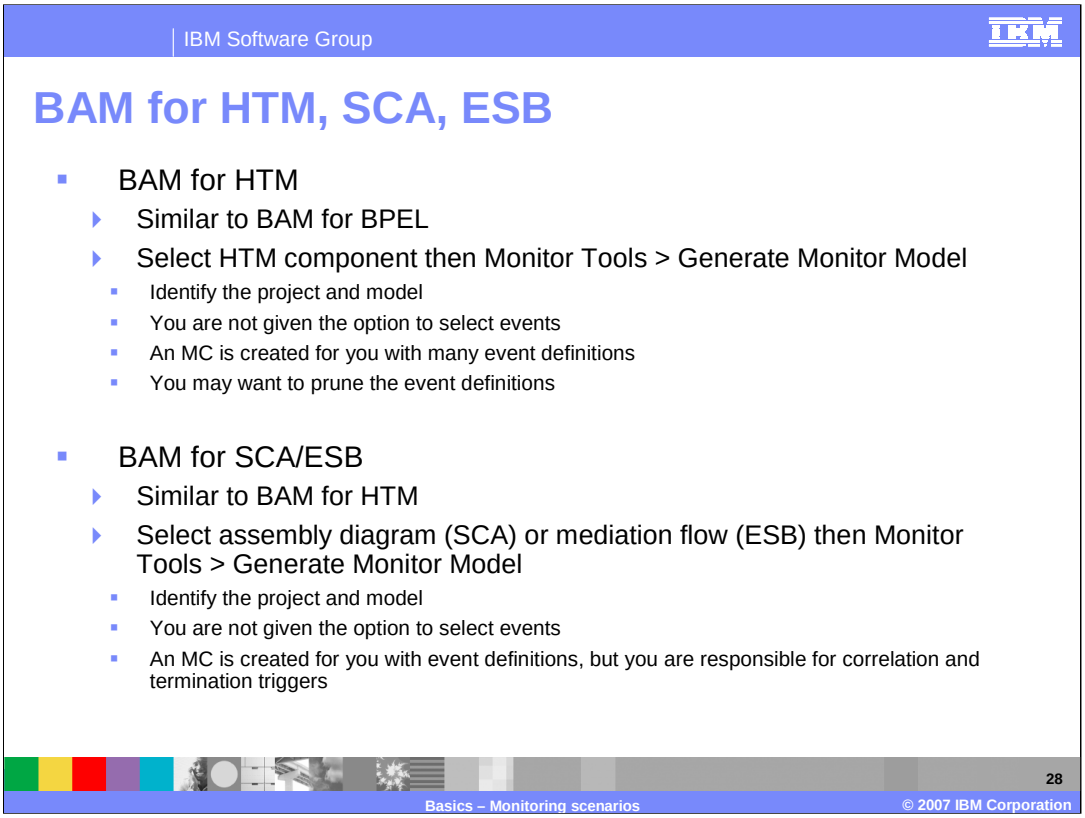
BPEL Construct \ Event Nature	ASSIGNED	CHANGED	COMPENSATED	COMPENSATING	COMPFAILED	CONDFALSE	CONDTRUE	CORRELATION	CREATED	DEASSIGNED	DELETE	DELETED	ENTRY	ESCALATED	EV_ESCALATED	EV_RECEIVED	EXIT	EXPIRED	FAILED	FAILING	FAULTSET	FCOMPLETED	FRETRIED	OUTPUTSET	RESTARTED	RESUMED	SKIPPED	STOPPED	SUSPENDED	TERMINATED	TERMINATING	VM_CREATED	VM_DELETED	VM_REFRESHED	VM_TRANSFERRED
Process			x	x	x			x				x	x		x	x	x		x	x					x	x			x	x	x	x	x		x
Scope			x	x	x								x		x	x	x		x	x							x			x					
While						x	x																				x								
Invoke													x				x	x	x			x	x				x	x		x					
Receive													x				x													x					
Pick													x				x	x	x								x								
Wait													x				x		x											x					
Staff (inline HT)	x								x	x				x			x	x	x		x	x	x	x			x	x		x		x	x	x	x
Sequence																											x								
Flow						x	x																				x								
Switch																											x								
Reply																	x		x								x								
Throw																	x		x								x								
Empty																	x										x								
Script (Snippet)																	x		x			x	x				x	x		x					
Assign																	x		x								x								
Rethrow																	x										x								
Compensate													x				x		x											x					
Terminate																	x										x								
variable		x																																	

It is imperative that you understand which events create and terminate monitoring contexts. If you do not select the proper event types to be emitted from your business process, then when you auto-generate the monitor model from the BPEL, you may end up missing some required event types. These event types are necessary to automatically manage the monitoring contexts in the monitor model. For example, the ENTRY nature is generally used to create monitoring contexts, and the EXIT nature is generally used terminate the monitoring contexts. So you should include these when selecting the events in the Event Monitor tab.

ECS IDs

- For **process** events:
  - The ECSCurrentID provides the ID of the process instance.
  - The ECSParentID is the ECSCurrentID before the process instance start event of the current process.
- For **activity** events:
  - The ECSCurrentID provides the ID of the activity.
  - The ECSParentID provides the ID of the containing process.
- For **scope** events:
  - The ECSCurrentID provides the ID of the scope.
  - The ECSParentID provides the ID of the containing process.
- For **link** events:
  - The ECSCurrentID provides the ID of the source activity of the link.
  - The ECSParentID provides the ID of the containing process

The context data in the common base event is used for correlation of events to monitoring context instances. Generally, the ECScurrentID is the identifier of a specific activity in the process, and the ECSParentID is the identifier of the parent process which contains the activity. These IDs are used for correlation when you auto-generate the monitor model, and you may need them yourself when specifying correlation for events that you use in the model.



The Monitor model created for a human task (HTM) is designed to monitor at the *instance level.*. This means that each instance of a human task will result in a separate monitoring context instance, since the key for each MC instance is the human task instance ID. When you auto-generate the model, it is similar to auto-generating for BPEL. But you are not given the option to select the event types, so you may want to go through the ones that are generated and prune them.

BAM for SCA and ESB is similar to the HTM scenario except for the initial WebSphere Integration Developer artifact that is selected before bringing up the context menu. For SCA, the assembly diagram within the module project is selected, while for ESB, a mediation flow is selected. Note also, that a monitoring context will be auto-generated for you but you will have to setup the correlation and termination criteria yourself.

References

- DB2 Replication Redbook
  http://www.redbooks.ibm.com/redbooks/pdfs/sg246828.pdf
- XPath 2.0, http://www.w3.org/TR/xpath20/
- Information center on BPEL/HTM event structure and format
  http://publib.boulder.ibm.com/infocenter/dmndhelp/v6rxmx/topic/com.ibm.wsps.mon.doc/doc/cmon_BPC_events.html
- WebSphere Business Integration Server Foundation 5.1 and CEI RedPaper
  http://www.redbooks.ibm.com/abstracts/redp3915.html
- Monitor Web site for samples and best practices
  http://www.ibm.com/software/integration/wbimonitor/library/tutorials.html

This slide shows some references including for DB2 replication, XPath, BPEL Common Base Event formats and CEI. The CEI red paper is based on Server Foundation 5.1 but is useful to understand the CEI APIs. Also, a Web site is available that contains Monitor samples and best practices.

Summary

- You reviewed various modeling scenarios including general BAM and BAM for BPEL

In this presentation you have reviewed monitor modeling scenarios, including general Business Activity Monitoring and BAM for BPEL.

Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject= Feedback about WBMonitorV602_BasicsScenarios.ppt

You can help improve the quality of IBM Education Assistant content by providing feedback.

Trademarks, copyrights, and disclaimers

The following terms are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both:

WebSphere

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Product data has been reviewed for accuracy as of the date of initial publication. Product data is subject to change without notice. This document could include technical inaccuracies or typographical errors. IBM may make improvements or changes in the products or programs described herein at any time without notice. Any statements regarding IBM's future direction and intent are subject to change or withdrawal without notice, and represent goals and objectives only. References in this document to IBM products, programs, or services does not imply that IBM intends to make such products, programs or services available in all countries in which IBM operates or does business. Any reference to an IBM Program Product in this document is not intended to state or imply that only that program product may be used. Any functionally equivalent program, that does not infringe IBM's intellectual property rights, may be used instead.

Information is provided "AS IS" without warranty of any kind. THE INFORMATION PROVIDED IN THIS DOCUMENT IS DISTRIBUTED "AS IS" WITHOUT ANY WARRANTY, EITHER EXPRESS OR IMPLIED. IBM EXPRESSLY DISCLAIMS ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NONINFRINGEMENT. IBM shall have no responsibility to update this information. IBM products are warranted, if at all, according to the terms and conditions of the agreements (for example, IBM Customer Agreement, Statement of Limited Warranty, International Program License Agreement, etc.) under which they are provided. Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products in connection with this publication and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products.

IBM makes no representations or warranties, express or implied, regarding non-IBM products and services.

The provision of the information contained herein is not intended to, and does not, grant any right or license under any IBM patents or copyrights. Inquiries regarding patent or copyright licenses should be made, in writing, to:

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

Performance is based on measurements and projections using standard IBM benchmarks in a controlled environment. All customer examples described are presented as illustrations of how those customers have used IBM products and the results they may have achieved. The actual throughput or performance that any user will experience will vary depending upon considerations such as the amount of multiprogramming in the user's job stream, the I/O configuration, the storage configuration, and the workload processed. Therefore, no assurance can be given that an individual user will achieve throughput or performance improvements equivalent to the ratios stated here.