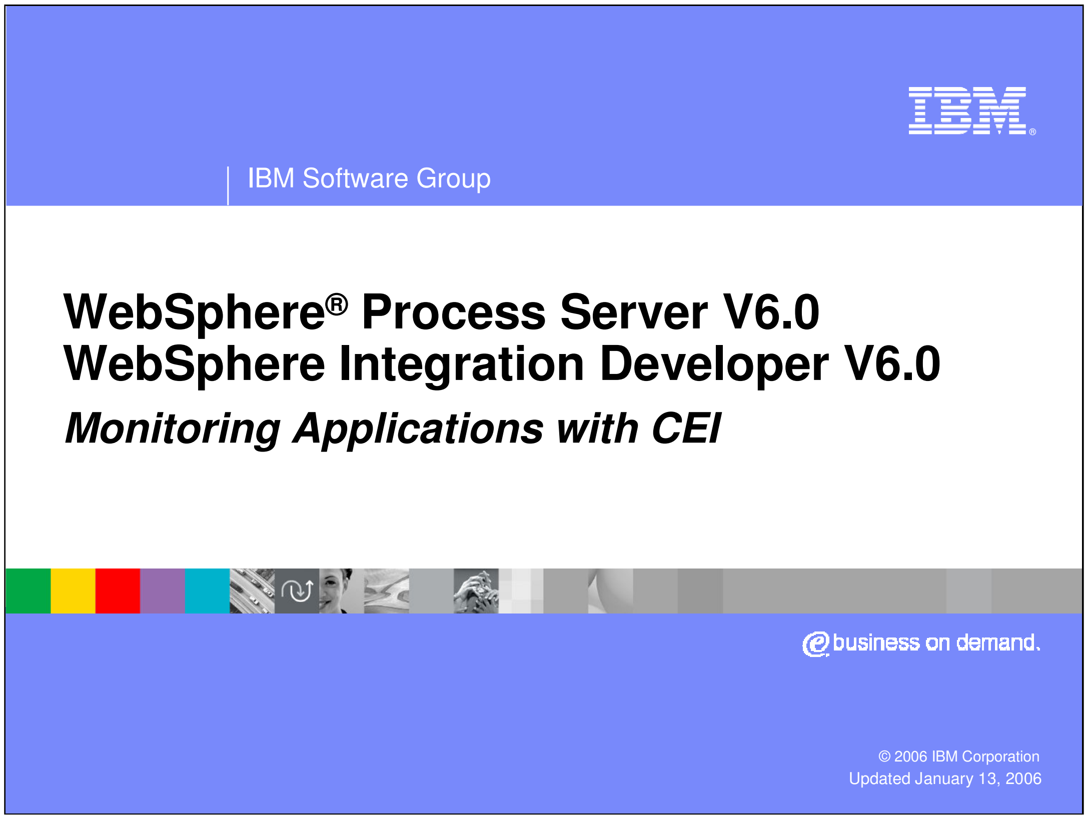


IBM Software Group

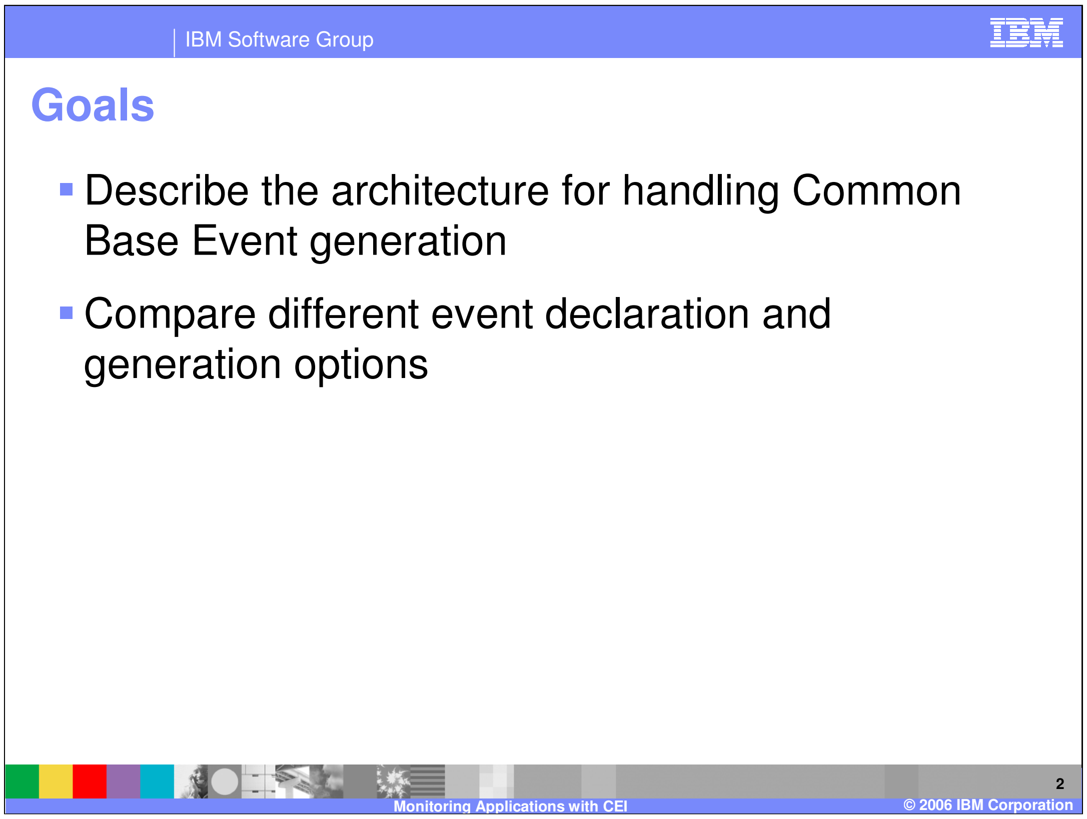
WebSphere® Process Server V6.0 WebSphere Integration Developer V6.0

Monitoring Applications with CEI

business on demand.

Updated January 13, 2006

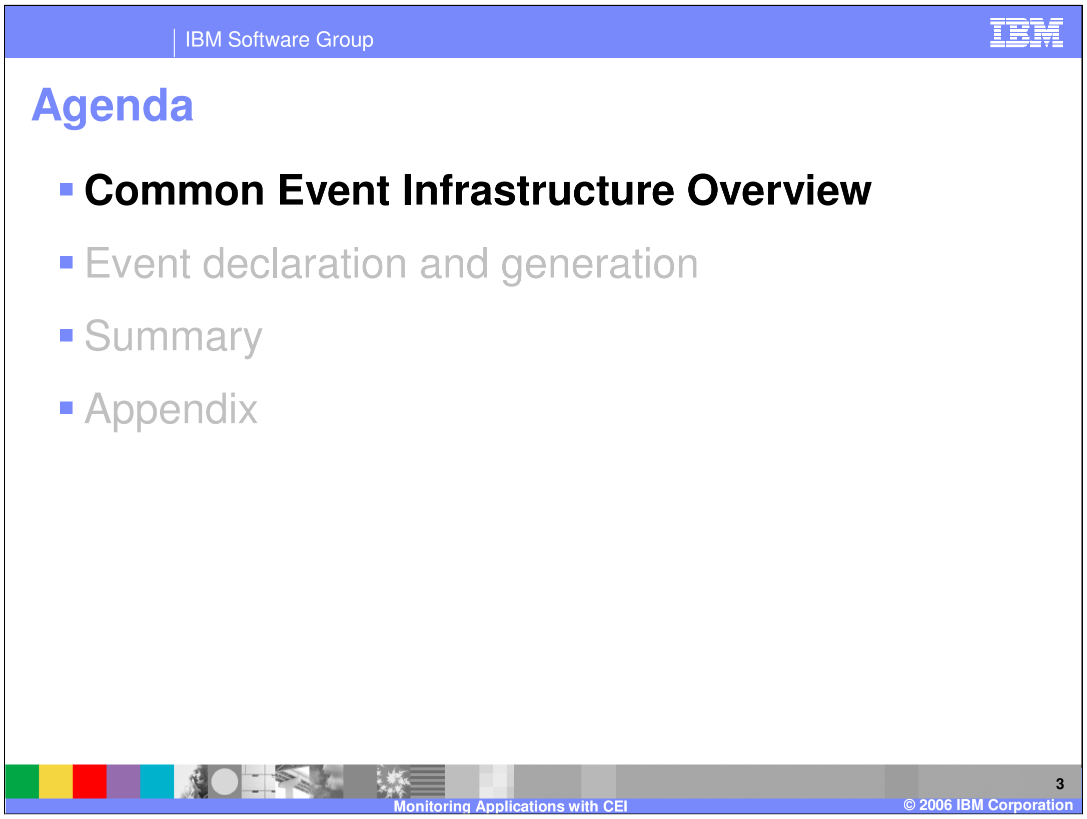
This presentation will focus on monitoring applications with the Common Event Infrastructure (CEI) in WebSphere Process Server and WebSphere Integration Developer V6.0.



Goals

- Describe the architecture for handling Common Base Event generation
- Compare different event declaration and generation options

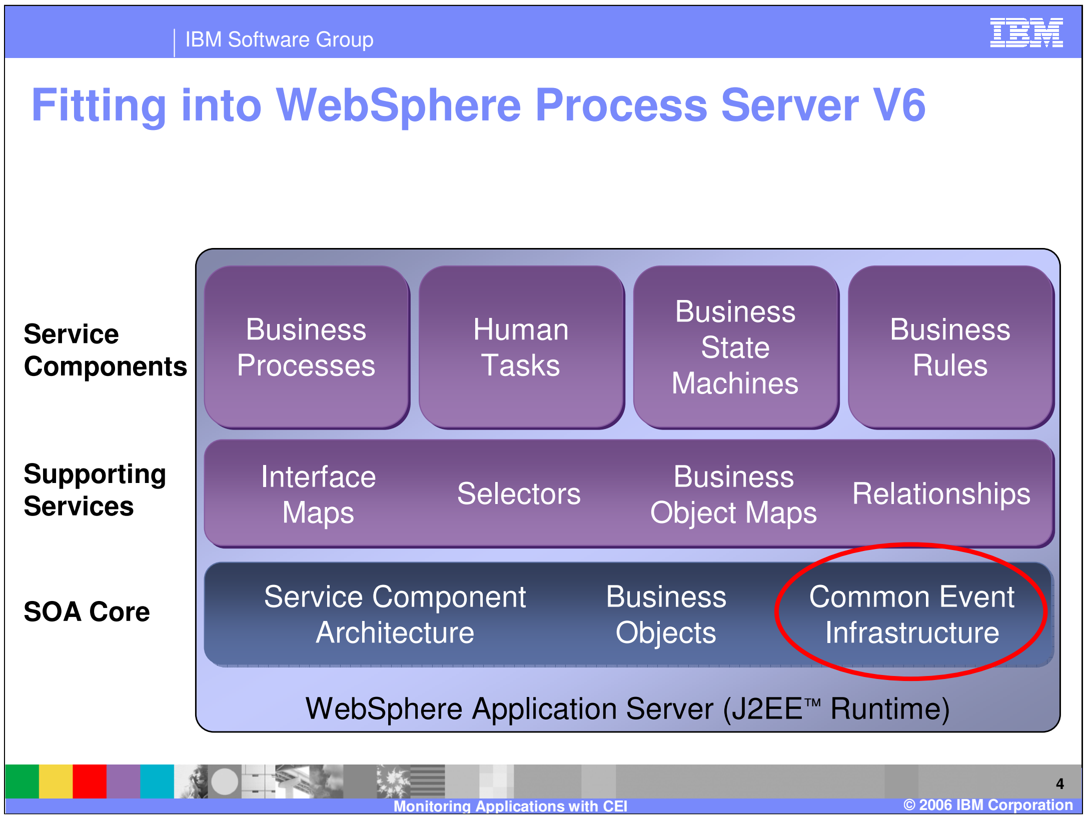
The goals of this presentation are to discuss the architecture for handling Common Base Event generation, event declaration and generation mechanisms, and clients that can be used to view and work with these events.



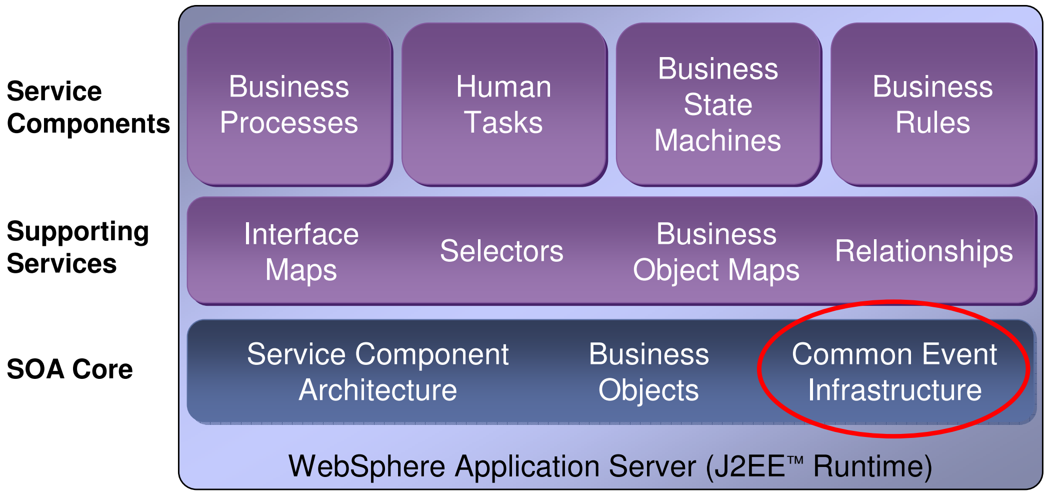
Agenda

- **Common Event Infrastructure Overview**
- Event declaration and generation
- Summary
- Appendix

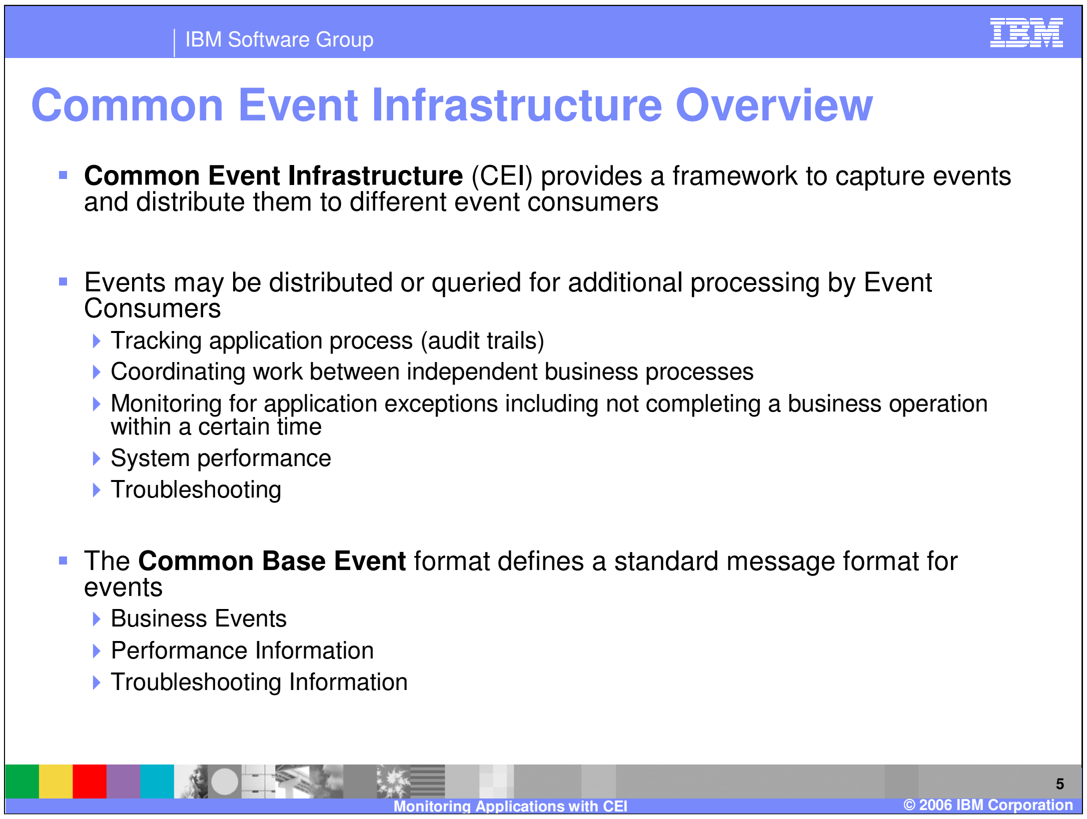
This section will provide an overview of the Common Event Infrastructure (CEI).



Fitting into WebSphere Process Server V6



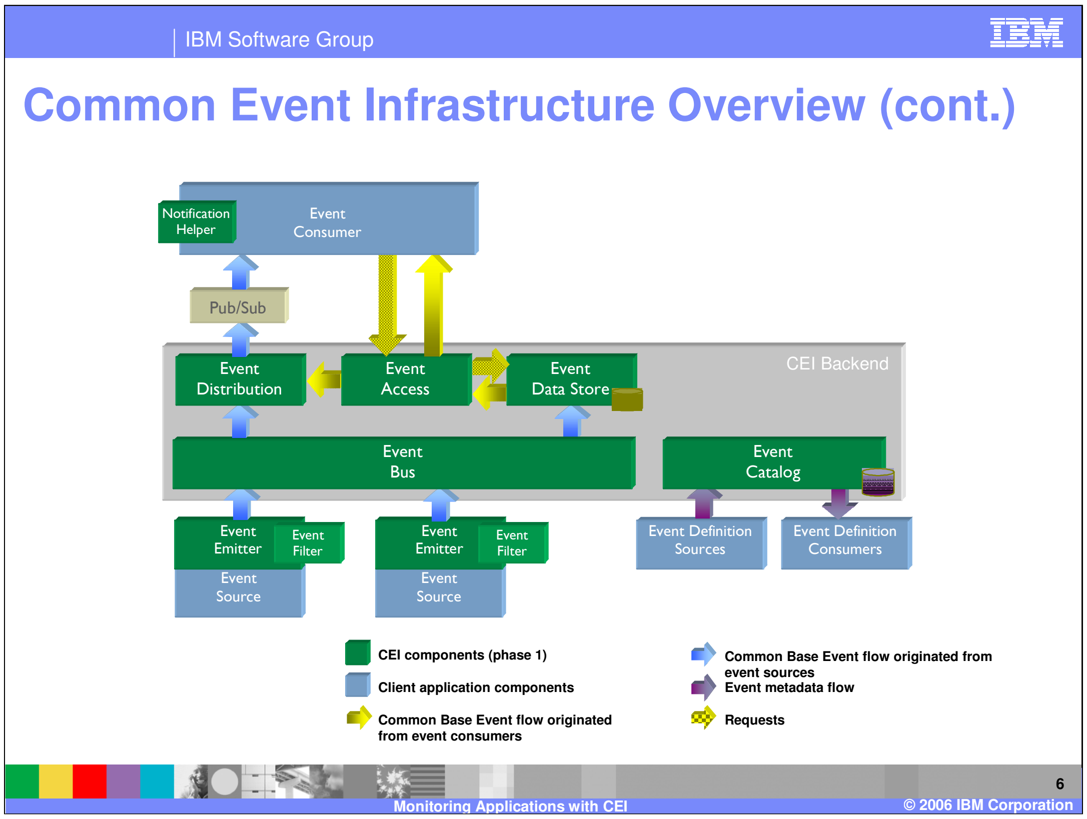
The common event infrastructure or CEI is one of the SOA core components that are a part of WebSphere Process Server. CEI is available to all the supporting services and service components that are running in the process server environment. All of these supporting services and service components have the ability to generate events that can be handled by CEI and then handled by various clients such as those provided by Tivoli® and WebSphere Business Monitor. These events include administrative or management events or application related events that you might want to act upon within your application. The event definition is being standardized through the OASIS standards body – so that other companies as well as customers can use the same infrastructure to monitor through-out their environments.



Common Event Infrastructure Overview

- **Common Event Infrastructure** (CEI) provides a framework to capture events and distribute them to different event consumers
- Events may be distributed or queried for additional processing by Event Consumers
  - Tracking application process (audit trails)
  - Coordinating work between independent business processes
  - Monitoring for application exceptions including not completing a business operation within a certain time
  - System performance
  - Troubleshooting
- The **Common Base Event** format defines a standard message format for events
  - Business Events
  - Performance Information
  - Troubleshooting Information

CEI is a built-in framework for capturing events from various sources and then distributing them to event consumers. CEI is the IBM implementation for handling events that could be generated for a variety of purposes. Events can be used for administrative tasks such as setting up audit trails within an application or for performance information or troubleshooting purposes. The types of events handled by the CEI must follow the Common Base Event format, which is a standard format that IBM has helped to define with other industry vendors. This standard contains the details about what types of information must be in an event, such as a header, certain defined information, and a payload, which is more or less arbitrary and can include business, error or troubleshooting information.



IBM Software Group

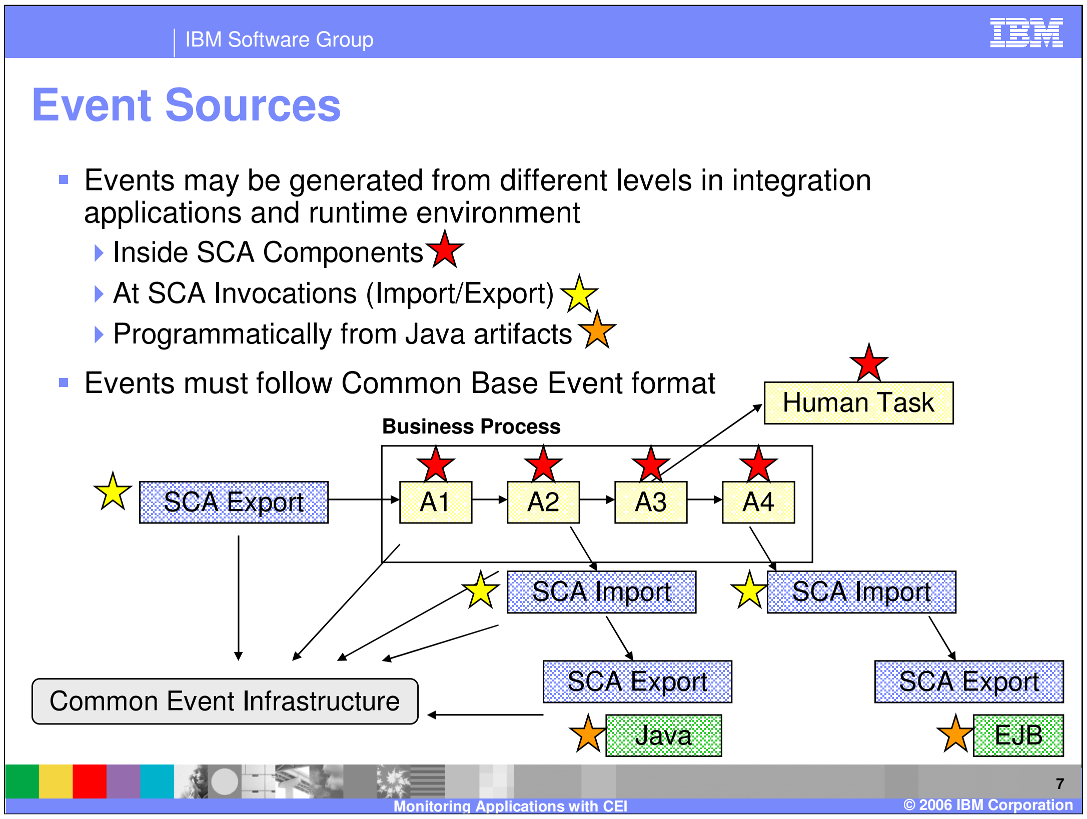
Common Event Infrastructure Overview (cont.)

6

Monitoring Applications with CEI

© 2006 IBM Corporation

Here is an overall view of the CEI architecture. Events are accepted from a variety of event sources and can be filtered prior to being placed on the event bus. Events are then published to event consumers. Events can be stored locally in a datastore or published on a topic or other messaging destination, where message consumers can wait for them and act upon them. Part of the framework includes APIs that consumers can use to access events that are stored locally. Event publishing and storing can be enabled either simultaneously or individually. The event catalog is another persistent store and is part of the CEI framework. It contains information about the format of the events that are being published. The event catalog is used by event consumers to work with the events that they receive or retrieve from the database. Because the Common Base Event standard leaves the payload definition open and arbitrary, there must be some type of indicator of what type of information is contained in an event, such as the format and how to extract it. There are also specific APIs that event consumers can call to access that information. Schema information is not used at runtime for event generation.


IBM Software Group

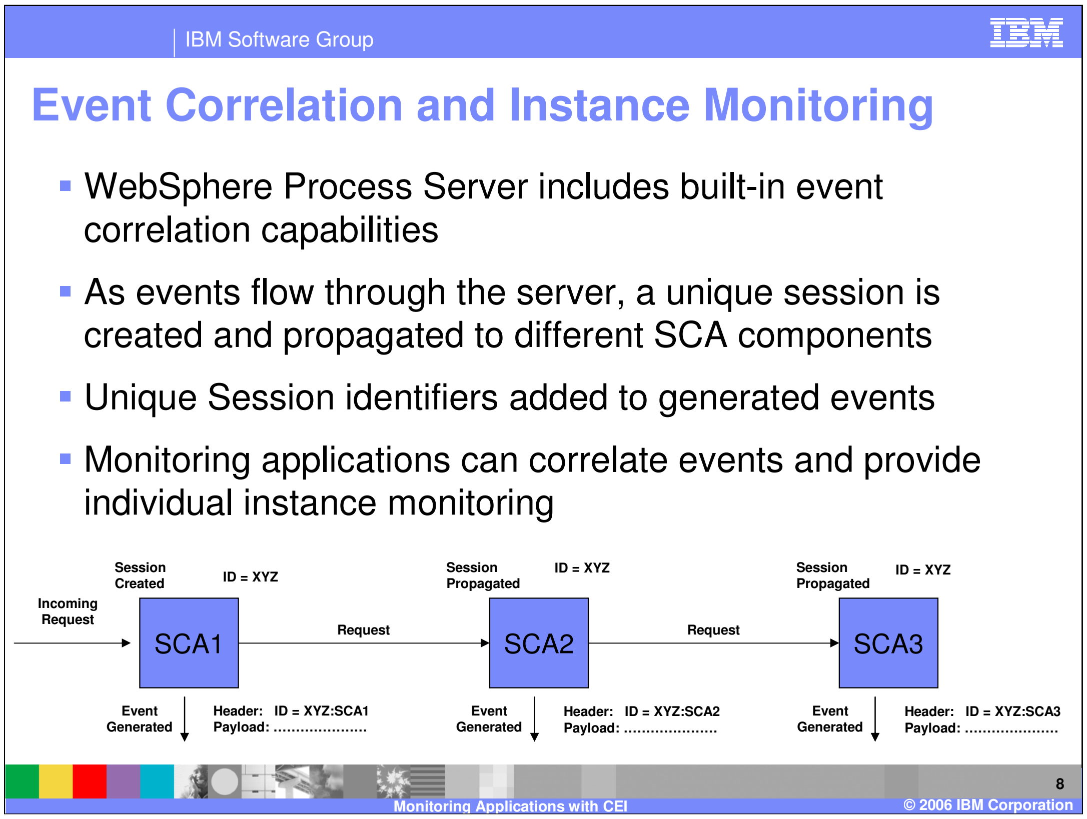
Event Sources

- Events may be generated from different levels in integration applications and runtime environment
  - Inside SCA Components
  - At SCA Invocations (Import/Export)
  - Programmatically from Java artifacts
- Events must follow Common Base Event format

Human Task
Business Process
SCA Export
A1
A2
A3
A4
SCA Import
SCA Import
SCA Export
SCA Export
Common Event Infrastructure
Java
EJB

7
Monitoring Applications with CEI
© 2006 IBM Corporation

Events can be generated from a variety of event sources within the process server environment. This can done inside SCA components, marked on this slide with a red flag. On individual components, such as human tasks, you can set up specific events or Common Base Events to be generated within the business process for each of the different activities, allowing for a fine level of granularity from within a component. You can also enable events to be generated at the SCA level, so any type of invocation of a specific interface on an SCA component can have an event that is fired and can also be caught from the event infrastructure. Finally, for more specific situations where a very customized event containing very specific business information is needed, you can create your own events and place this business information into the Common Base Event payload. These events can also be captured by CEI. CEI is a central event infrastructure available for monitoring applications and the progress of business transactions as well as for performing application event notification and handling. The event producer, such as a Java class or EJB session bean, might be firing an event which is consumed by a consumer that would react to that event. Event generation is not limited to administration and monitoring capabilities.



IBM Software Group

Event Correlation and Instance Monitoring

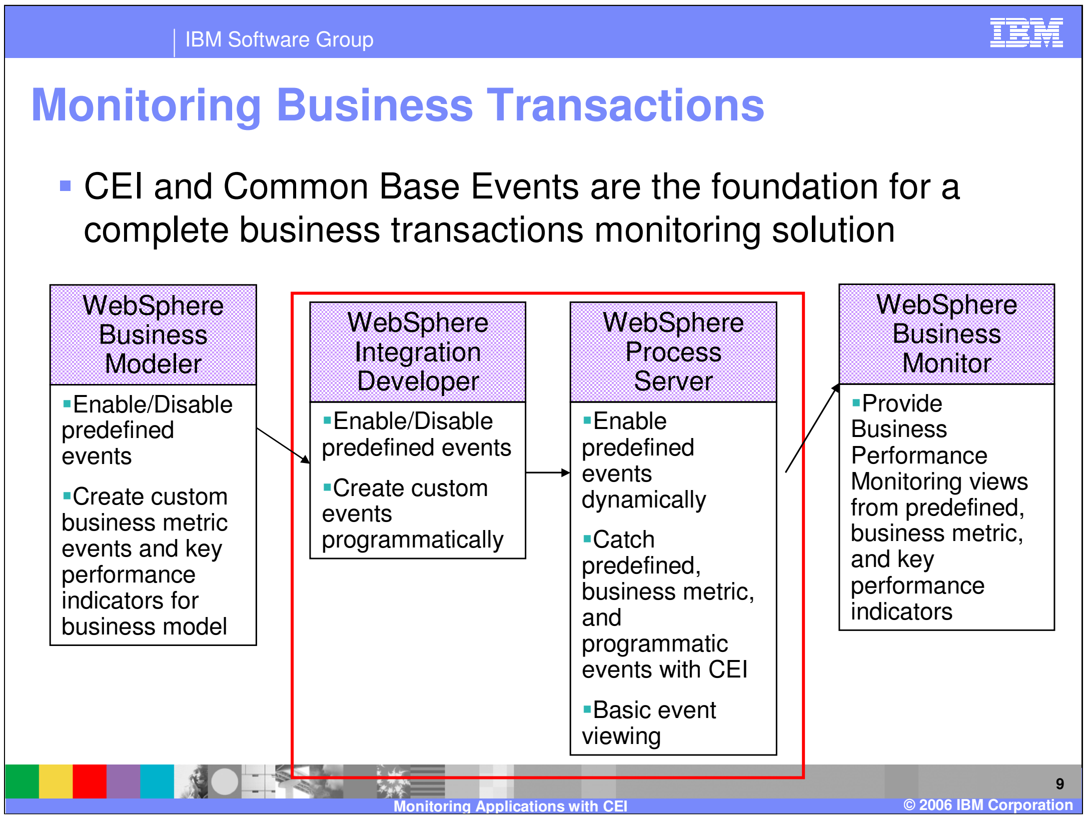
- WebSphere Process Server includes built-in event correlation capabilities
- As events flow through the server, a unique session is created and propagated to different SCA components
- Unique Session identifiers added to generated events
- Monitoring applications can correlate events and provide individual instance monitoring

8

Monitoring Applications with CEI

© 2006 IBM Corporation

Anyone who has tried to create their own even handling framework knows that the biggest challenge is correlation between different events that are generated in different transactions only seconds apart or days apart. How do you determine if the message that was just received is related to one received two days ago? How do you know that it is part of the same business transaction? Strategy to accomplish this usually involves using application specific information in the event such as an order number or some other unique identifier, or extending these operations to include a unique identifier specifically for event correlation. This approach is very labor intensive for most situations. With the process server environment, event correlation is provided for you and built in as part of the SCA framework. Anytime an SCA component is called, a session ID is created and can be used to established a correlation ID to the Common Base Events that are generated. For example, if a request comes in to SCA component 1 there is a session established, a context created, and a unique ID generated for this transaction. When that event or any event coming out of that SCA component is generated, that session ID will be used as well as that particular SCA component ID, making up the unique identifier that can be used as part of the correlation and specified in the generated event. As the event continues on from one SCA component to another the session is propagated and appended with the SCA component ID as part of the header information on any events generated to the CEI framework. This information is placed in the header of the event rather than the payload to prevent loss of the session data. An event consumer can then use the header to create specific views or timelines for transactions.


IBM Software Group

Monitoring Business Transactions

- CEI and Common Base Events are the foundation for a complete business transactions monitoring solution

WebSphere Business Modeler
▪Enable/Disable predefined events
▪Create custom business metric events and key performance indicators for business model
WebSphere Integration Developer
▪Enable/Disable predefined events
▪Create custom events programmatically
WebSphere Process Server
▪Enable predefined events dynamically
▪Catch predefined, business metric, and programmatic events with CEI
▪Basic event viewing
WebSphere Business Monitor
▪Provide Business Performance Monitoring views from predefined, business metric, and key performance indicators

9

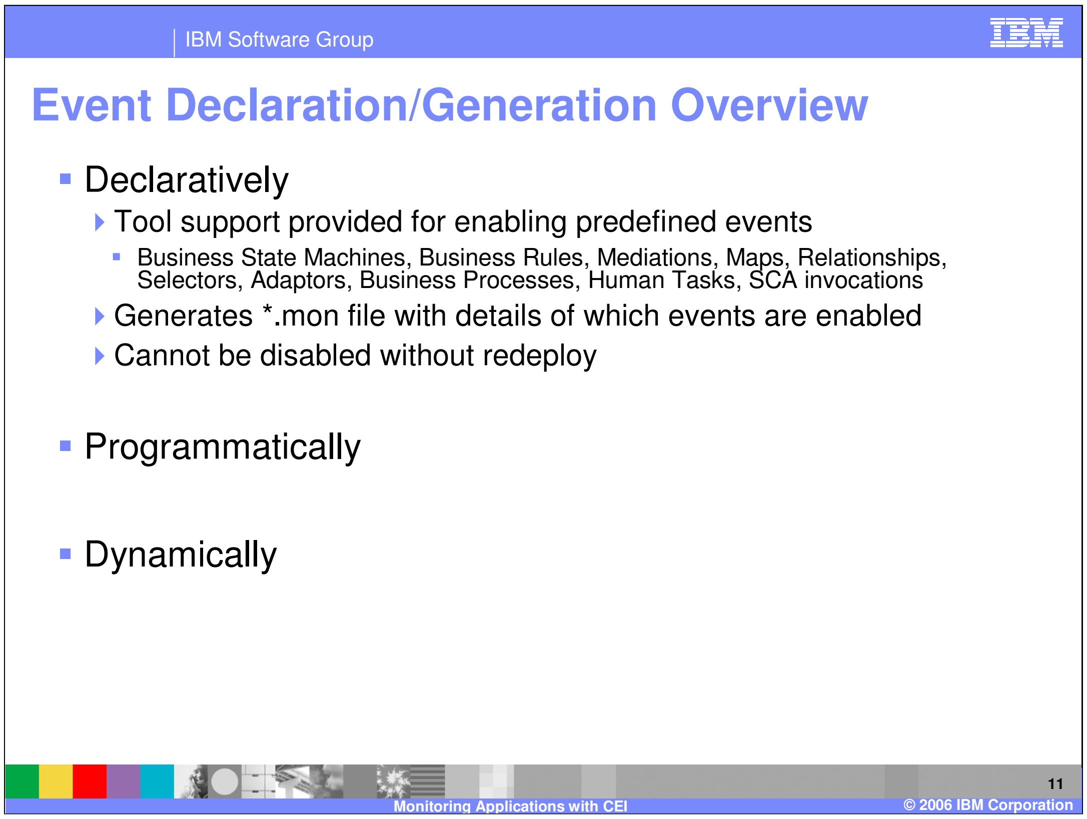
Monitoring Applications with CEI © 2006 IBM Corporation

While this presentation will only focus on WebSphere Integration Developer and WebSphere Process Server, they are just part of the overall business transaction monitoring strategy. This strategy becomes stronger with WebSphere Business Monitor and WebSphere Business Modeler, which provide additional capabilities for defining and monitoring events. This slide attempts to outline the roles each of these products plays in this overall strategy. The Modeler allows you to enable and disable predefined events and create custom business metric events which are more specific to your application, such as a customer order.   You can also declare key performance indicators which will look at aggregated values and how the results are matching expected outcomes. WebSphere Integration Developer also provides the ability to enable and disable predefined events within components or at the SCA level. You can also create custom events programmatically using the Java APIs for Common Base Event and CEI. The Process Server has the main framework for catching these predefined events that you enabled in Business Modeler or Integration Developer or for business model events or custom programmatic events in your application. Event notification can also be enabled dynamically for specific components, which can be used by administrators for problem determination or informational purposes. There is also a basic event consumer or event viewer included with the Process Server. The WebSphere Business Monitor is meant for industrial strength viewing of business metrics and key performance indicators and provides the ability to work with events and their correlation IDs, put them together and give you an overall view of your business transaction, showing key performance indicators such as meeting or exceeding thresholds.

Agenda

- Common Event Infrastructure Overview
- **Event declaration and generation**
- Summary

This section will cover event declaration and generation.



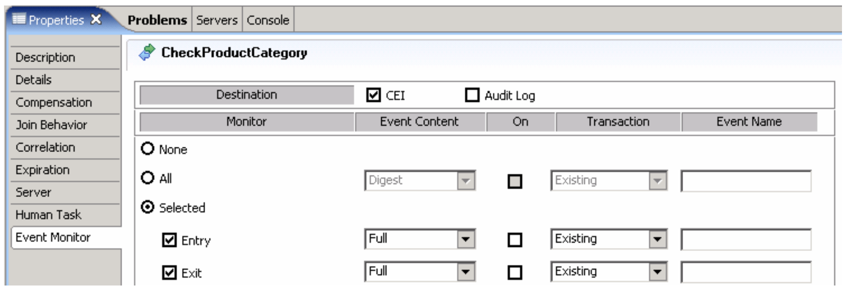
Event Declaration/Generation Overview

- Declaratively
  - Tool support provided for enabling predefined events
    - Business State Machines, Business Rules, Mediations, Maps, Relationships, Selectors, Adaptors, Business Processes, Human Tasks, SCA invocations
  - Generates *.mon file with details of which events are enabled
  - Cannot be disabled without redeploy
- Programmatically
- Dynamically

There are three ways to declare events that will be generated. This can be done with the WebSphere Integration Developer. You can enable specific events to be generated for the business state machines, business rules, mediations, maps, relationships, selectors, and all the different SCA components. When you enable events for the different components inside the development environment, there a is .mon file generated. You will not see this file unless you go to the Physical Files view or the navigator view but it is generated and used at run time to determine whether an event should be generated at a specific point. The events that are set up within WebSphere Integration Developer can only be changed in the development environment. They become part of the deployed application and there is no way to disable them without editing them in the development environment and redeploying. These events become part of the business process that was developed by an integration specialist; an administrator should not have the capability to disable them. Events can also be declared and generated programmatically in Java. Finally, there is an administrative option to generate more events dynamically at run time, if it is necessary to get more information about a particular component.

Declaratively Enabled Predefined Events

- Event Monitor tab available on different elements of a component to enable event generation
- Select Events
  - All, Entry, Exit, and so on
- Select Level
  - Empty, Full, Digest
- Select Transaction behavior
  - Existing (Same), New, None

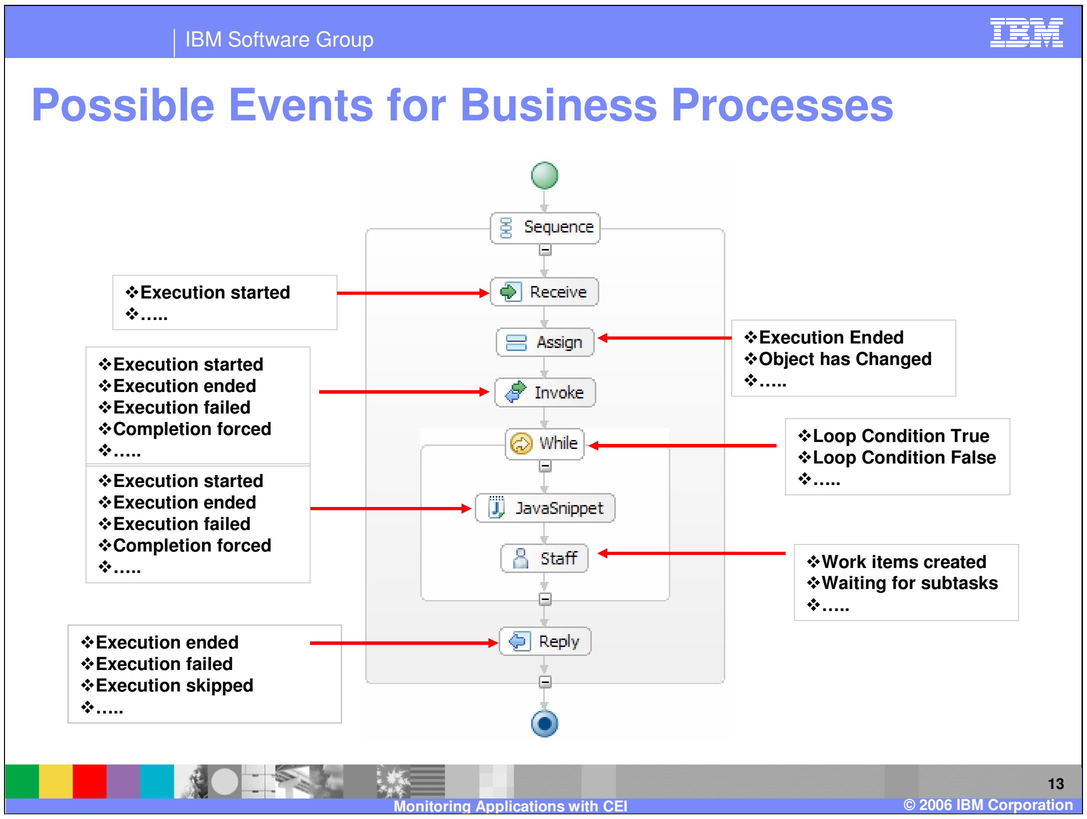


CheckProductCategory

Destination: [✓ CEI] Audit Log

Monitor	Event Content	On	Transaction	Event Name
None				
All	Digest	☐	Existing	
[✓ Selected]				
✓ Entry	Full	☐	Existing	
✓ Exit	Full	☐	Existing	

To enable predefined events for most components, select the component either in its own editor or within the assembly editor for SCA level event generation and there will be an event tab in the properties view. Select the properties tab and you will be able to specify the number of events you want to generate for different states. There is an ALL option, or SELECTED option, which provides a finer level of granularity on what events are generated. The event context can be EMTPY, DIGEST, or FULL with FULL containing the most information. In addition to selecting the checkbox, you must explicitly enable event generation. This maintains the state and the level of event that might have been generated in the past without losing that setting. If you are using BPEL processes or human tasks you will see a destination setting because they can generate events to the CEI infrastructure. Audit log capabilities are part of these components. This audit log setting allows events to be generated and stored in the BPE database. Use the CEI setting because this framework is available across all components and it provides a single point for event consumers to work from.



IBM Software Group

Possible Events for Business Processes

❖Execution started
❖.....

❖Execution Ended
❖Object has Changed
❖.....

❖Execution started
❖Execution ended
❖Execution failed
❖Completion forced
❖.....

❖Loop Condition True
❖Loop Condition False
❖.....

❖Execution started
❖Execution ended
❖Execution failed
❖Completion forced
❖.....

❖Work items created
❖Waiting for subtasks
❖.....

❖Execution ended
❖Execution failed
❖Execution skipped
❖.....

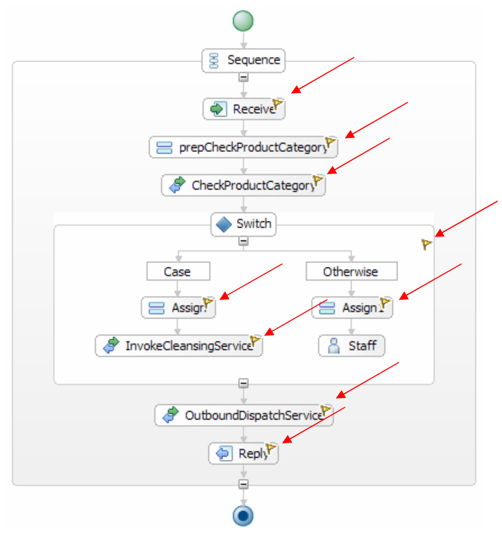
13

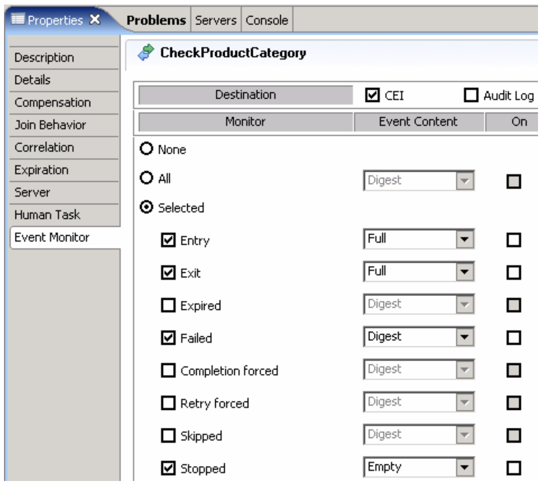
Monitoring Applications with CEI

© 2006 IBM Corporation

This slide shows some of the events that could be enabled in a BPEL process. This list is not exhaustive. It is intended to give you an idea of the types of events that you could enable and event consumers could then look for. For example, Looking at the timestamps for an “Execution started” in a receive activity, and an “Execution Ended” in a Reply, would allow you to determine how long the business process took. Using a “started” and “ended” on an inline human task activity would allow you to determine how long it takes for people to complete work items for the task.

Events Indications in Business Processes





CheckProductCategory

Destination ☑ CEI ☐ Audit Log

Monitor	Event Content	On
○ None		
○ All	Digest	☐
● Selected		
☑ Entry	Full	☐
☑ Exit	Full	☐
☐ Expired	Digest	☐
☑ Failed	Digest	☐
☐ Completion forced	Digest	☐
☐ Retry forced	Digest	☐
☐ Skipped	Digest	☐
☑ Stopped	Empty	☐

14

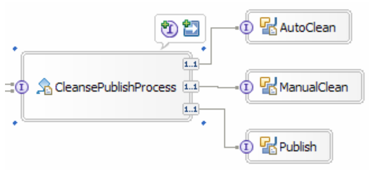
Monitoring Applications with CEI

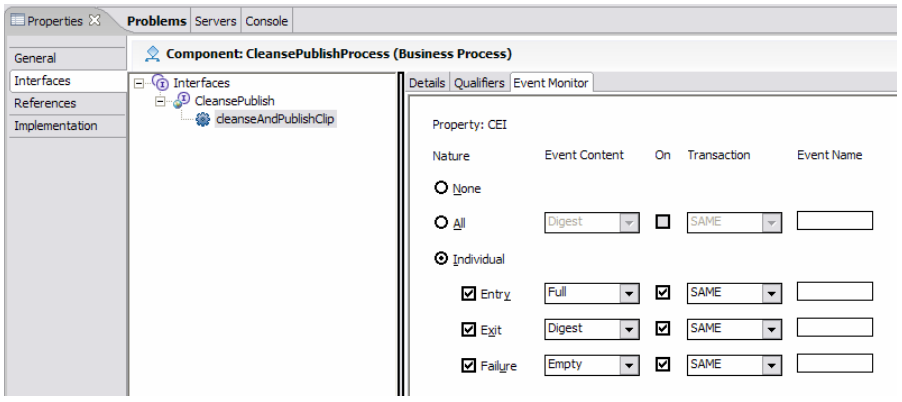
© 2006 IBM Corporation

When you enable events within the components, there are yellow flag icon indicators shown in the editors signifying event generation has been set up on the component. These indicators are used in the different component editors such as human task, business rule group, selector, and business state machines, to signal event generation has been added.

Declaratively Enabled SCA Invocation Events

- Specified on interface of SCA component, Export or Import
- Select specific events and payload levels



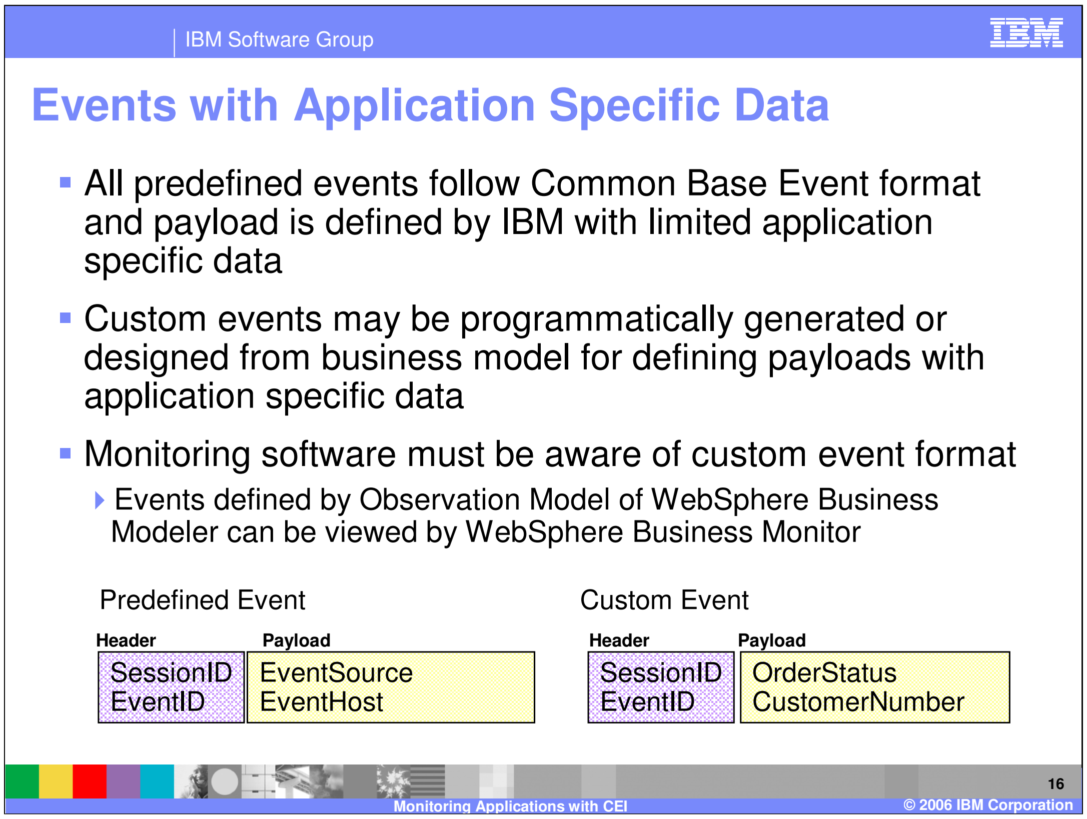


Component: CleansePublishProcess (Business Process)

Property: CEI

Nature	Event Content	On	Transaction	Event Name
None				
All	Digest	☐	SAME	
[✓ Individual]				
☑ Entry	Full	☑	SAME	
☑ Exit	Digest	☑	SAME	
☑ Failure	Empty	☑	SAME	

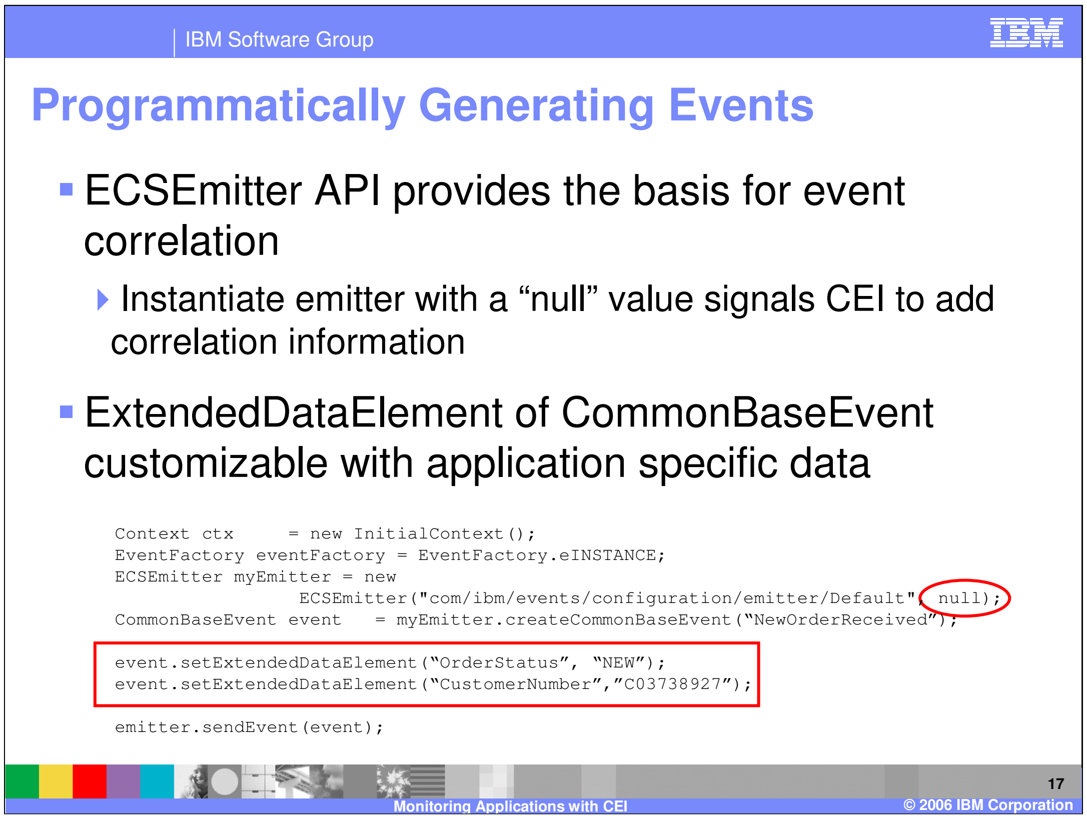
Event generation at the SCA level is enabled in the assembly editor by selecting a particular component and then selecting interfaces. On the Event Monitor tab, you can specify events that can be generated when this component is invoked. An event could be generated on entry for a component, when the component ends, or if there is a failure with the invocation of the component. At the low level invocation level, you can see which components are called and have a means of tracking coarse grained component invocation.



Events with Application Specific Data

- All predefined events follow Common Base Event format and payload is defined by IBM with limited application specific data
- Custom events may be programmatically generated or designed from business model for defining payloads with application specific data
- Monitoring software must be aware of custom event format
  - Events defined by Observation Model of WebSphere Business Modeler can be viewed by WebSphere Business Monitor

The amount of application-specific data that you will get from predefined events enabled in the component, or at the SCA level, is limited to details about the component. It will not include much (if any) application-specific data. If a business object is involved with the state change at the time the event is generated, it will be included in the event. To get more of this type of data, you will need to define customized events, either through the WebSphere Business Modeler product or programmatically, using Java. For example, in the predefined events, you will find the event source and possibly the event host, whereas a custom event would produce more business-specific information such as order status, or customer number.



IBM Software Group

Programmatically Generating Events

- ECSEmitter API provides the basis for event correlation
  - Instantiate emitter with a “null” value signals CEI to add correlation information
- ExtendedDataElement of CommonBaseEvent customizable with application specific data

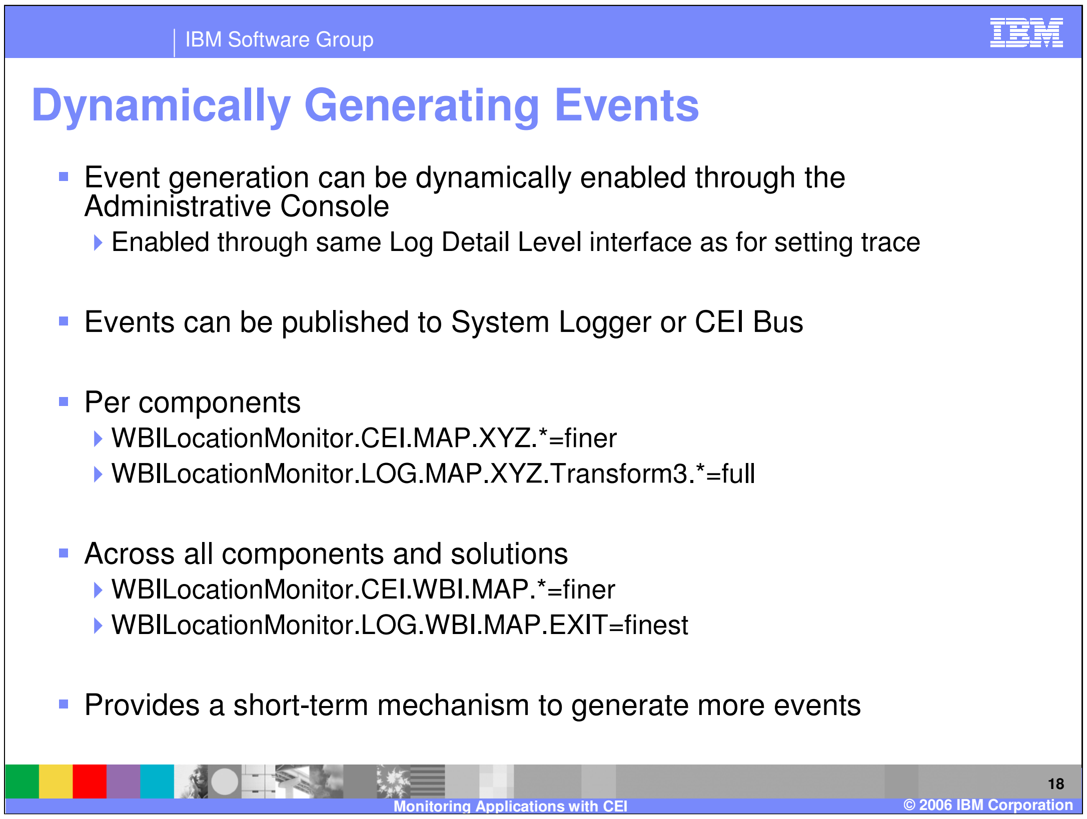
```
Context ctx      = new InitialContext();
EventFactory eventFactory = EventFactory.eINSTANCE;
ECSEmitter myEmitter = new
                 ECSEmitter("com/ibm/events/configuration/emitter/Default", null);
CommonBaseEvent event    = myEmitter.createCommonBaseEvent(“NewOrderReceived”);

event.setExtendedDataElement(“OrderStatus”, “NEW”);
event.setExtendedDataElement(“CustomerNumber”,”C03738927”);

emitter.sendEvent(event);
```

17

Monitoring Applications with CEI

© 2006 IBM Corporation

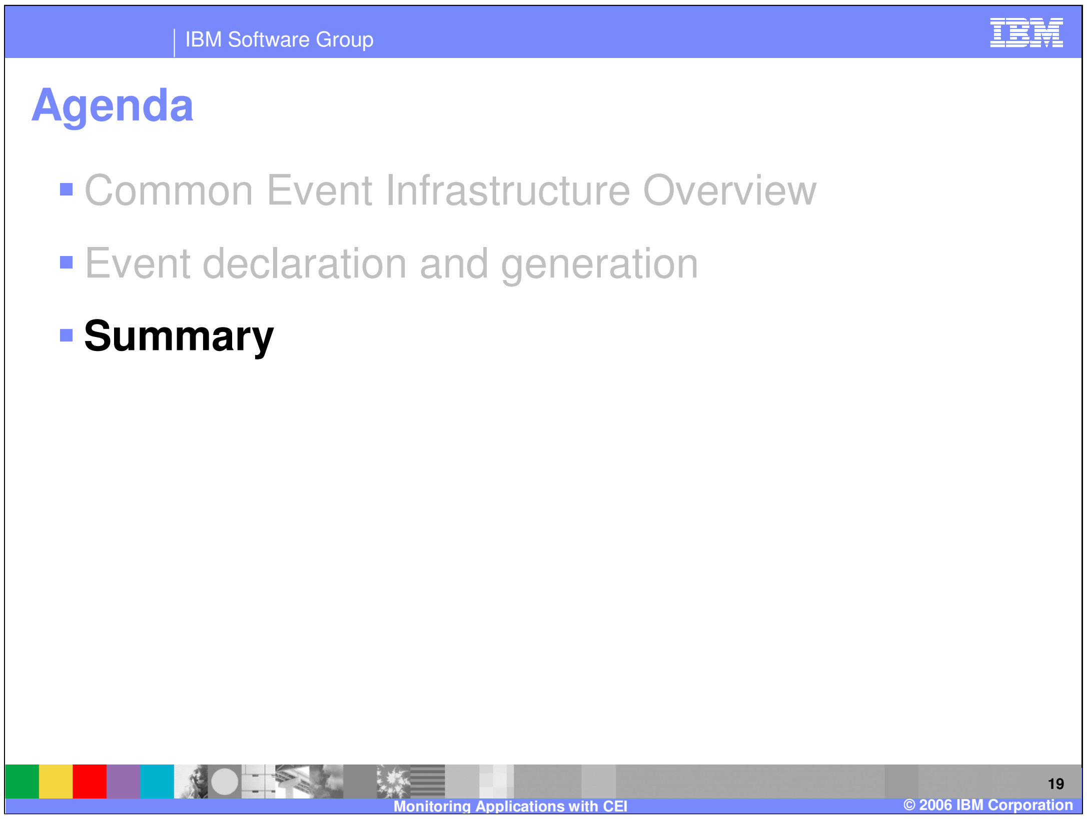
The emitter class has been extended to include a new ECS emitter that can be used within your Java code and by instantiating an object of that type and passing in a value of null. This is an indication to the runtime that it should use the existing SCA context. By using the session ID in that context as part of the event generation, events can participate in the correlation. Other than a different API, everything else remains the same. You still put your application-specific information in some type of extended data element, in this example OrderStatus or CustomerNumber, and the event can then be sent through the emitter.



Dynamically Generating Events

- Event generation can be dynamically enabled through the Administrative Console
  - Enabled through same Log Detail Level interface as for setting trace
- Events can be published to System Logger or CEI Bus
- Per components
  - WBILocationMonitor.CEI.MAP.XYZ.*=finer
  - WBILocationMonitor.LOG.MAP.XYZ.Transform3.*=full
- Across all components and solutions
  - WBILocationMonitor.CEI.WBI.MAP.*=finer
  - WBILocationMonitor.LOG.WBI.MAP.EXIT=finest
- Provides a short-term mechanism to generate more events

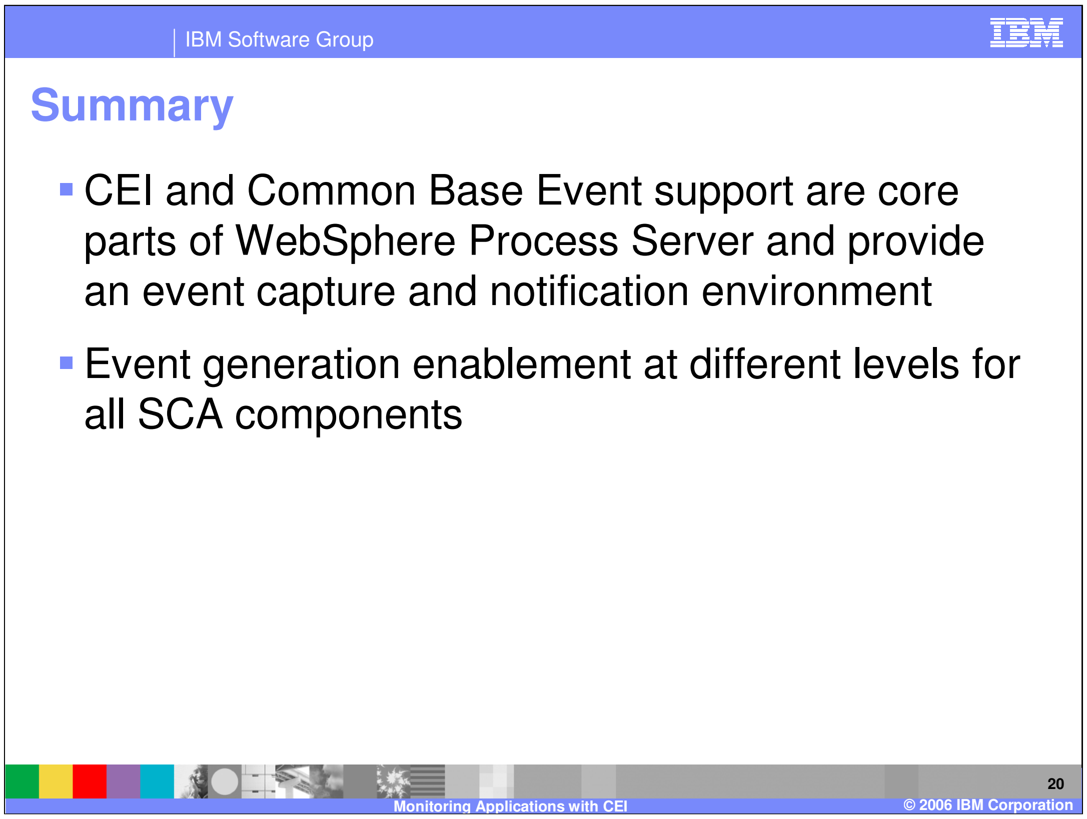
The third method of generating events is dynamically. This can be done by an administrator, using the administrative console. If you did not enable specific event information for a particular component and are seeing some unexpected results such as business transactions or human tasks taking longer than expected, you can enable event generation for a short period of time for the component. These events can be published to the CEI bus and persisted or published to other event consumers. The events could also be saved to a log file. You can set up varying levels of granularity to determine what type of event you want to generate. Events can be generated at a per component level or for specific components or specific component parts such as MAP.XYZ. Different levels of content can be specified for the generated events. These dynamic event settings are independent of the setting specified within WebSphere Integration Developer. If you have set up specific event generation, you cannot reduce the amount of event generation using this mechanism. If the level of event generation specified in the tool is higher, it will take precedence. This is designed to prevent an administrator from disabling event generation that was specified during development for business performance monitoring reasons. Dynamic event generation is configured in the same manner as for configuring trace, using the log and trace interface of the administrative console. This option is a good for short term event enablement.



Agenda

- Common Event Infrastructure Overview
- Event declaration and generation
- **Summary**

This section will provide a summary of the CEI monitoring capability.



IBM Software Group

Summary

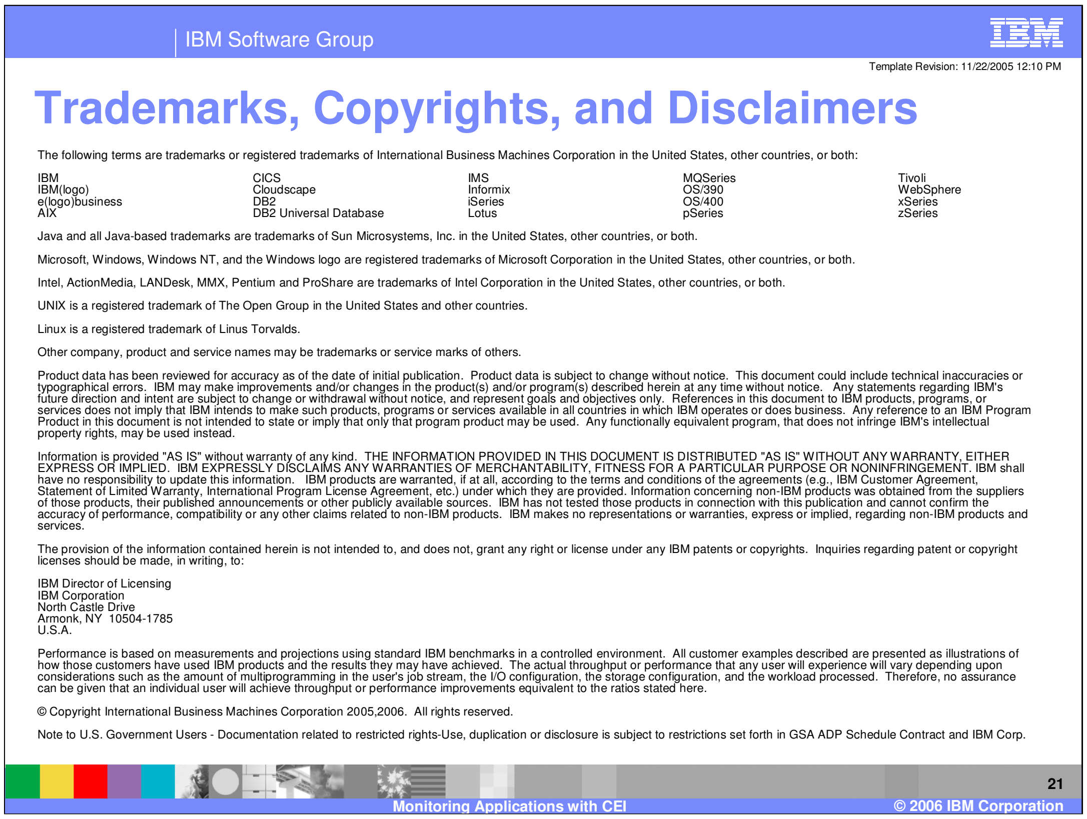
- CEI and Common Base Event support are core parts of WebSphere Process Server and provide an event capture and notification environment
- Event generation enablement at different levels for all SCA components

20

Monitoring Applications with CEI

© 2006 IBM Corporation

In summary, CEI monitoring is part of the SCA core components that make up WebSphere Process Server. SCA components that you generate can utilize this framework for generating business information. There is correlation support available for assisting in grouping events for the viewing of an end to end business transaction from separate events. You can specify events at many different levels, down to a very fine level of granularity for specific monitoring.



Trademarks, Copyrights, and Disclaimers