IBM Software Group

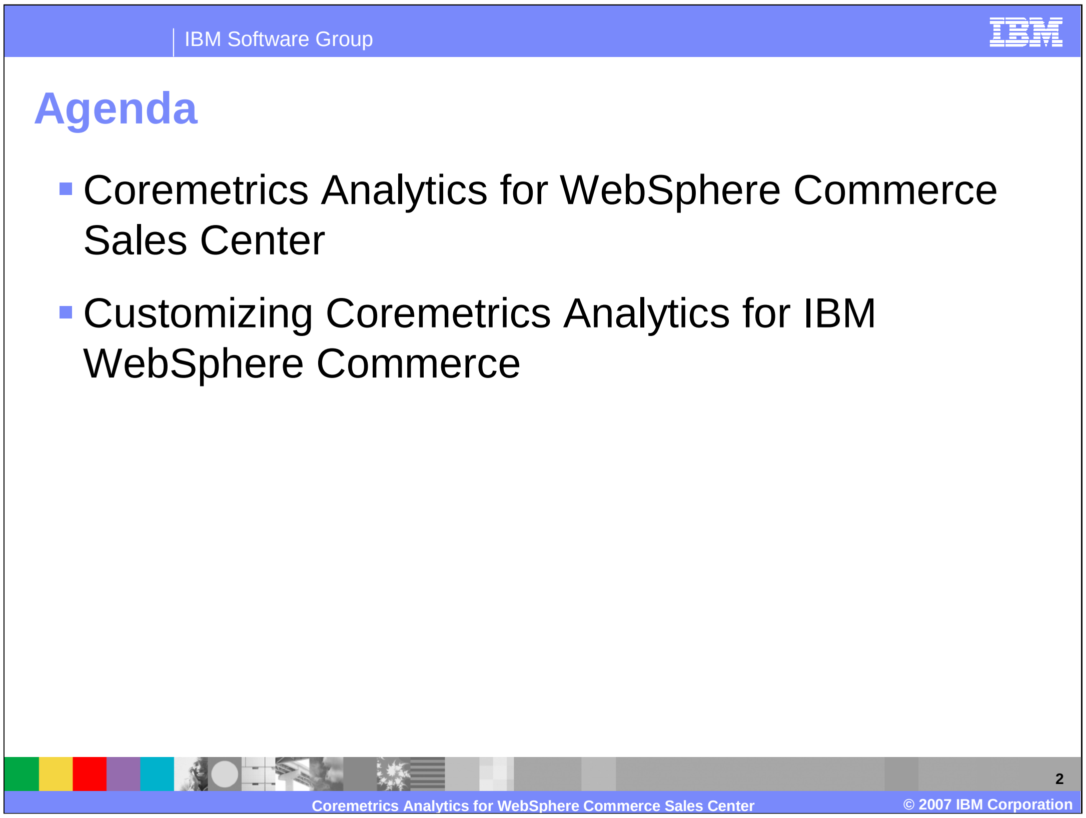
WebSphere® Commerce Feature Pack 2

Coremetrics Analytics for WebSphere Commerce Sales Center

business on demand.

Updated May 21, 2007

This presentation will discuss the Coremetrics Analytics for IBM WebSphere Commerce Sales Center enhancement within Feature Pack 2.



IBM Software Group

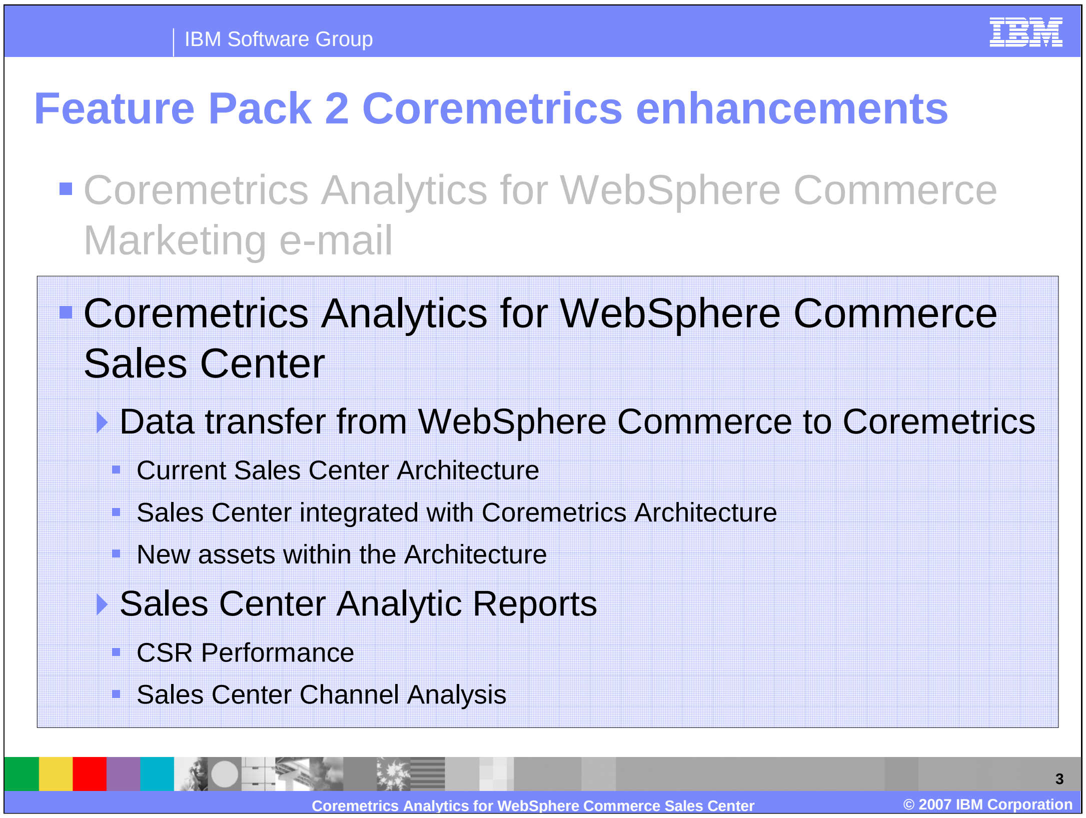
Agenda

- Coremetrics Analytics for WebSphere Commerce Sales Center
- Customizing Coremetrics Analytics for IBM WebSphere Commerce

2

Coremetrics Analytics for WebSphere Commerce Sales Center

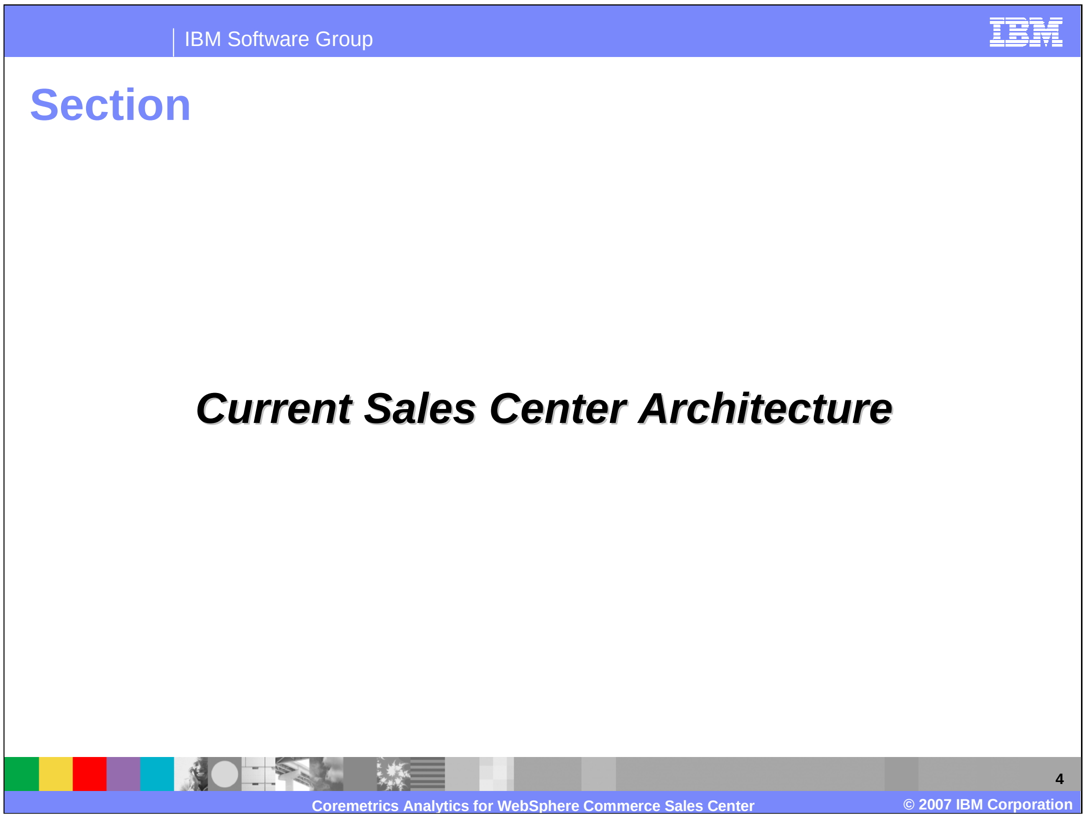
The agenda for this presentation will be to discuss the details of the Coremetrics Analytics for WebSphere Commerce Sales Center feature and what it takes to customize this feature.



Feature Pack 2 Coremetrics enhancements

- Coremetrics Analytics for WebSphere Commerce Marketing e-mail
- Coremetrics Analytics for WebSphere Commerce Sales Center
  - Data transfer from WebSphere Commerce to Coremetrics
    - Current Sales Center Architecture
    - Sales Center integrated with Coremetrics Architecture
    - New assets within the Architecture
  - Sales Center Analytic Reports
    - CSR Performance
    - Sales Center Channel Analysis

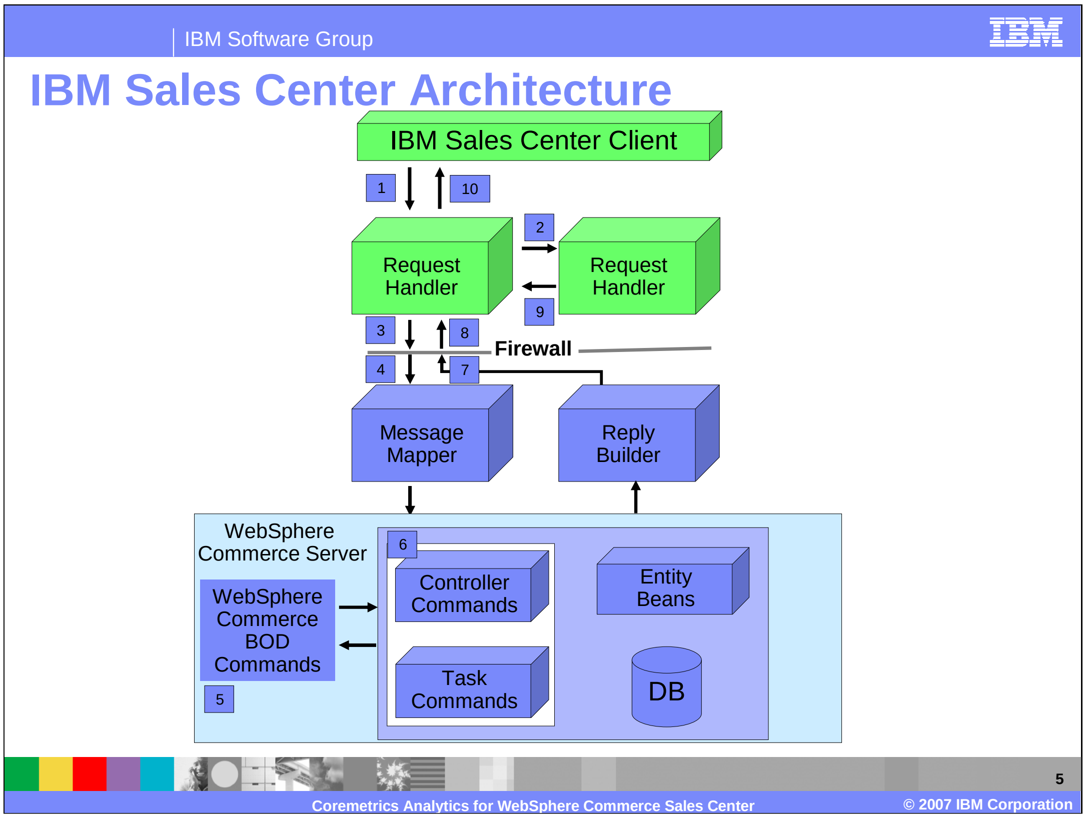
There are two main enhancements for WebSphere Commerce Feature Pack 2: Coremetrics Analytics for WebSphere Commerce Marketing e-mail and Coremetrics Analytics for WebSphere Commerce Sales Center. In this presentation, the latter will be discussed, specifically, data transfer from WebSphere Commerce to Coremetrics and Sales Center Analytic reports. You will look at the current Sales Center Architecture, how it integrates with Coremetrics and the new pieces within the architecture. Finally, the CSR Performance and the Sales Center Channel Analysis reports will be discussed.



Section

Current Sales Center Architecture

This section will discuss the current Sales Center Architecture.



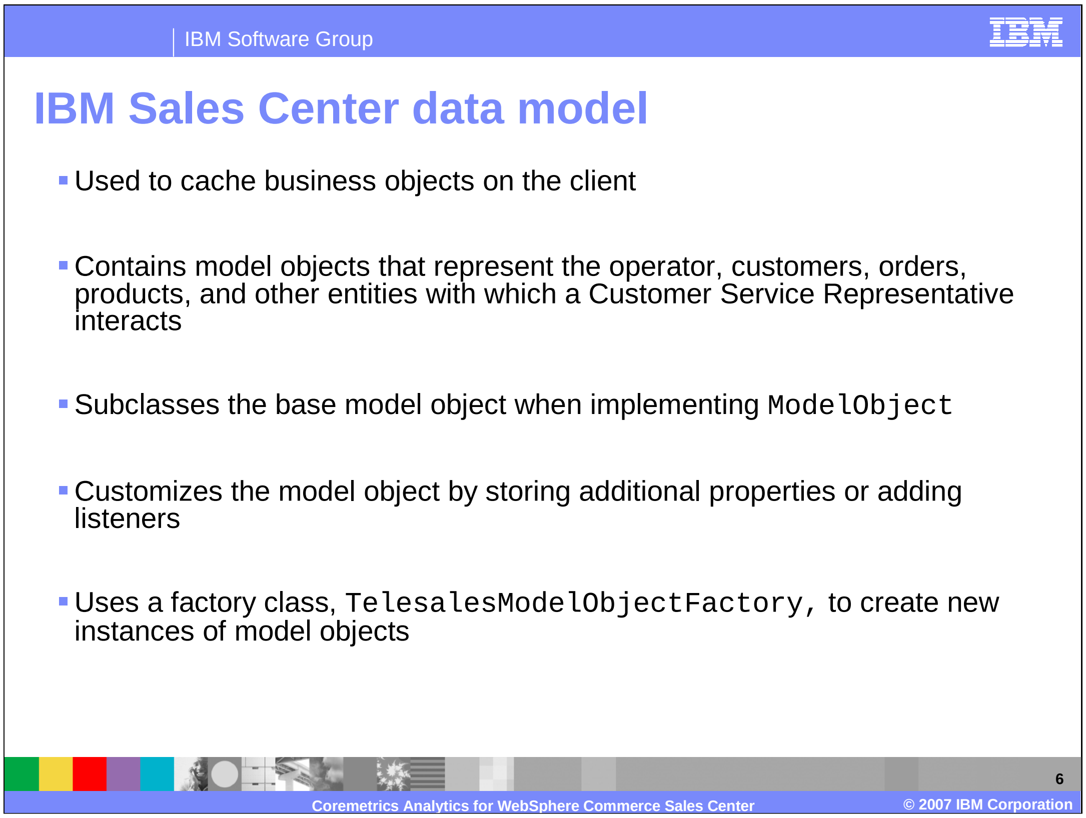
IBM Sales Center Architecture

The IBM Sales Center architecture, when viewed in its entirety, is comprised of the IBM Sales Center client, WebSphere Commerce Server, and a messaging architecture that allows for customization and extensibility.

IBM Sales Center uses this messaging architecture to communicate with the WebSphere Commerce Server. It is possible to customize the behavior of any of the three components of this architecture, thus allowing for changing the appearance or behavior of the client, modifying or creating messages and customizing and extending the server itself.

The following steps take place during the Sales Center request flow processing:

1. The IBM Sales Center client performs a service request.
2. The service request handler prepares a Business Object Document (BOD) message.
3. The message is sent from the client to the host machine.
4. The message mapper receives the message and maps the BOD to a WebSphere Commerce BOD command.
5. The WebSphere Commerce BOD command is invoked.
6. The WebSphere Commerce BOD command calls a WebSphere Commerce Controller command, which may call one or more task commands.
7. The reply, or response, builder constructs the response BOD.
8. The response is returned to the client machine.
9. The request handler receives and handles the response BOD.
10. The client user interface is updated on screen.



IBM Software Group

IBM Sales Center data model

- Used to cache business objects on the client
- Contains model objects that represent the operator, customers, orders, products, and other entities with which a Customer Service Representative interacts
- Subclasses the base model object when implementing `ModelObject`
- Customizes the model object by storing additional properties or adding listeners
- Uses a factory class, `TelesalesModelObjectFactory,` to create new instances of model objects

6

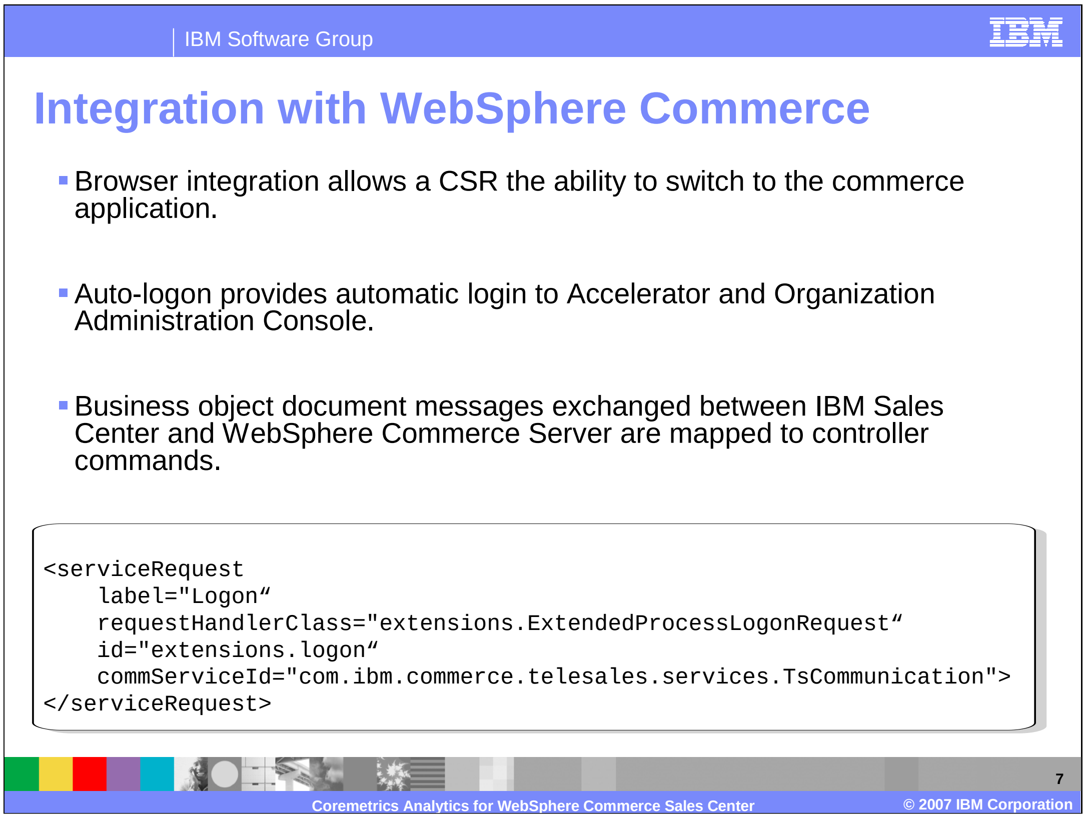
Coremetrics Analytics for WebSphere Commerce Sales Center

The data model is used to cache business objects on the client. The TelesalesModelManager class provides access to the model and contains convenience methods for accessing and updating child objects. The default IBM Sales Center data model contains model objects that represent the operator, customers, orders, products, and other commonly used objects. All of the objects found in the client cache are instances of ModelObject.

You can store additional properties in a model object by using the setData and getData methods. You can also add listeners to model objects that will be notified of any change to the model object. If a property is a list of model objects, use the ModelObjectList instead of a generic container object to ensure that property notification is properly handled.

There is a factory class, TelesalesModelObjectFactory, that should be used to create new instances of model objects. There is also a model object extension point that lets you register new model objects. This factory and extension provide support for replacing existing model object implementations with a new implementation. The new implementation must subclass the base model object.

Next, you will take a closer look at the specific customizations you can use to integrate IBM Sales Center and WebSphere Commerce.



IBM Software Group

IBM

Integration with WebSphere Commerce

- Browser integration allows a CSR the ability to switch to the commerce application.
- Auto-logon provides automatic login to Accelerator and Organization Administration Console.
- Business object document messages exchanged between IBM Sales Center and WebSphere Commerce Server are mapped to controller commands.

```
<serviceRequest
    label="Logon“
    requestHandlerClass="extensions.ExtendedProcessLogonRequest“
    id="extensions.logon“
    commServiceId="com.ibm.commerce.telesales.services.TsCommunication">
</serviceRequest>
```

7

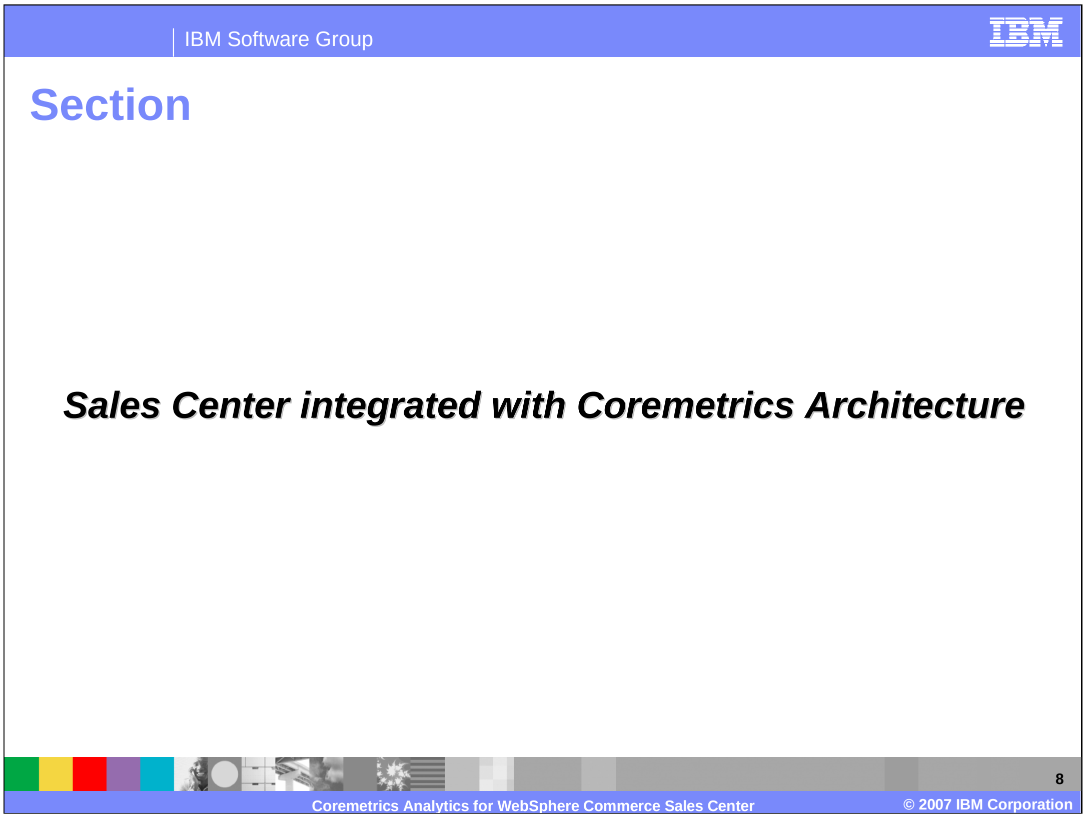
Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

If you integrate the IBM Sales Center application with an existing WebSphere Commerce application, you can leave part of the application browser interface intact and allow users to switch to the browser to interact with that portion of the application. Browser integration support is provided by declaring an extension to the WebSphere Everyplace Deployment extension point com.ibm.eswe.workbench.WctWebApplication. The browser integration in the IBM Sales Center launches the WebSphere Commerce Accelerator and the Organization Administration Console by default.

The IBM Sales Center client provides default auto-logon support for the WebSphere Commerce Accelerator and the WebSphere Commerce Organization Administration console. The auto-logon feature can be extended to any external Web-based application. The IBM Sales Center uses the browser application support provided by WebSphere Everyplace Deployment. The extension point that is used to define the IBM WebSphere Commerce Accelerator and IBM WebSphere Commerce Organization Administration Console applications is com.ibm.eswe.workbench.WctWebApplication.

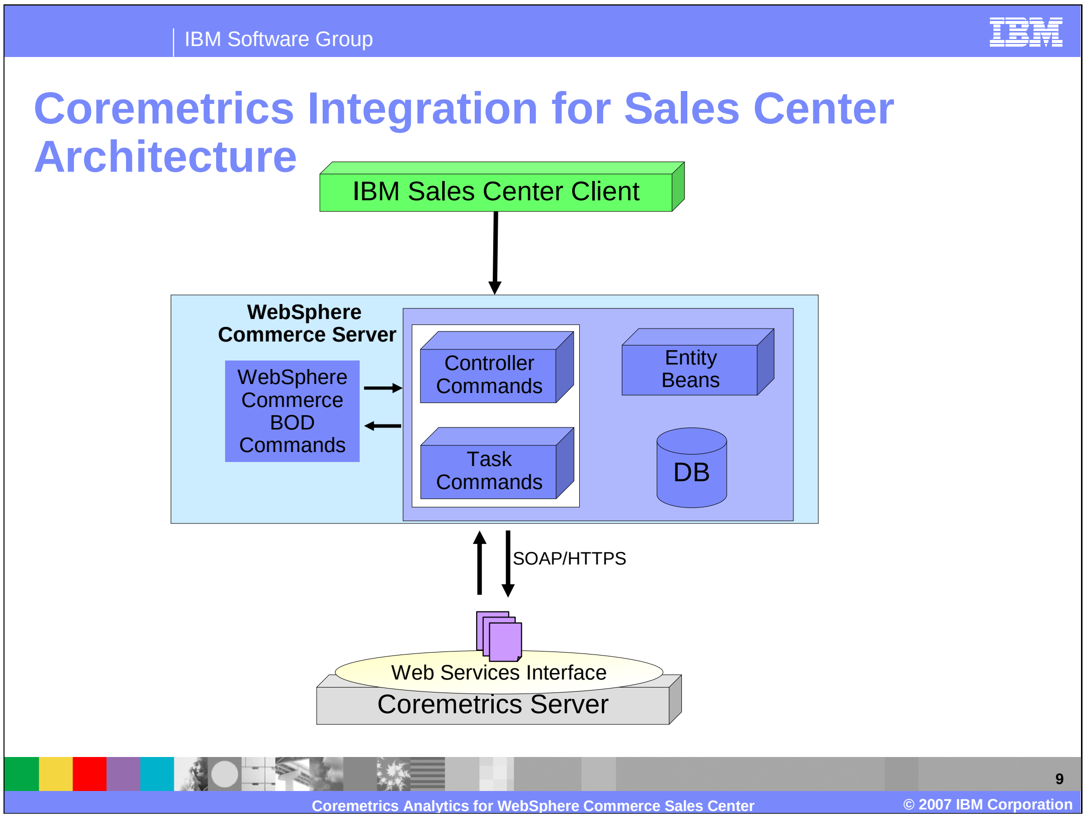
Business Object Document messages sent from the IBM Sales Center client to the WebSphere Commerce Server are mapped to WebSphere Commerce controller commands through the WebSphere Commerce message mapper. The WebSphere Commerce configuration file contains this message mapper entry. The BOD informs the receiving system what kind of message is in the data area, and status and error conditions. The BOD is formed of two parts, Noun and Verb. The Noun is a common business object. The actions performed on the Noun are the Verbs. BODs are designed to be extensible, while providing a common underlying architecture for integration. Changing the BOD format requires that you customize the service request handler.



Section

Sales Center integrated with Coremetrics Architecture

This section discusses how the Sales Center is integrated with Coremetrics.



IBM Software Group

Coremetrics Integration for Sales Center Architecture

IBM Sales Center Client

WebSphere Commerce Server

WebSphere Commerce BOD Commands

Controller Commands

Task Commands

Entity Beans

DB

SOAP/HTTPS

Web Services Interface

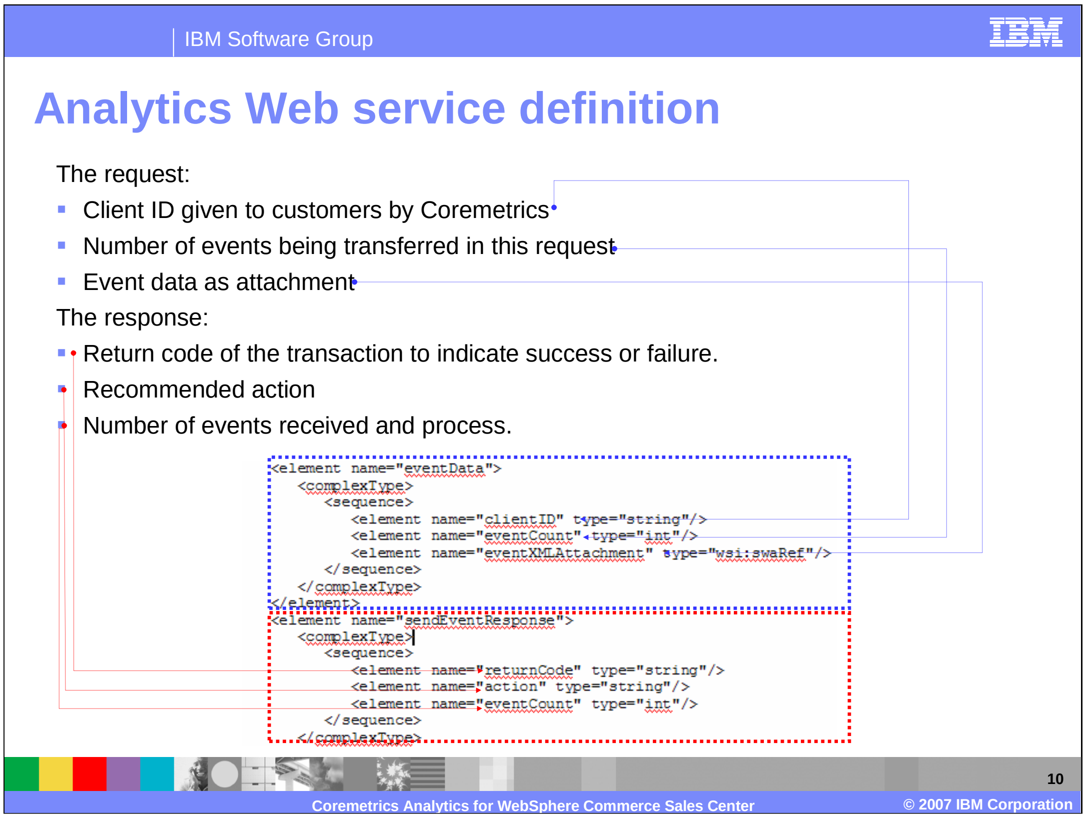
Coremetrics Server

9

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

Coremetrics is fundamentally a Web based analytics solution which collects data from a user’s browser. The traditional data collection strategy adopted by Coremetrics is a browser based solution where the information is passed to Coremetrics servers through JavaScript based image requests. Since Sales Center is a stand-alone application, Coremetrics JavaScript tagging will not work in this scenario and a different approach should be adopted for the transfer of information to Coremetrics. The function in this feature is based on the existing WebSphere Commerce business event infrastructure and the use of Web services to facilitate Sales Center Analytics with Coremetrics being a service provider.



Analytics Web service definition

The request:

- Client ID given to customers by Coremetrics
- Number of events being transferred in this request
- Event data as attachment

The response:

- Return code of the transaction to indicate success or failure.
- Recommended action
- Number of events received and process.

```
<element name="eventData">
   <complexType>
      <sequence>
         <element name="clientID" type="string"/>
         <element name="eventCount" type="int"/>
         <element name="eventXMLAttachment" type="wsi:swaRef"/>
      </sequence>
   </complexType>
</element>
<element name="sendEventResponse">
   <complexType>
      <sequence>
         <element name="returnCode" type="string"/>
         <element name="action" type="string"/>
         <element name="eventCount" type="int"/>
      </sequence>
   </complexType>
```

The request made by WebSphere Commerce will consist of three parts:

The client ID given to customers by Coremetrics

The number of events being transferred in this request

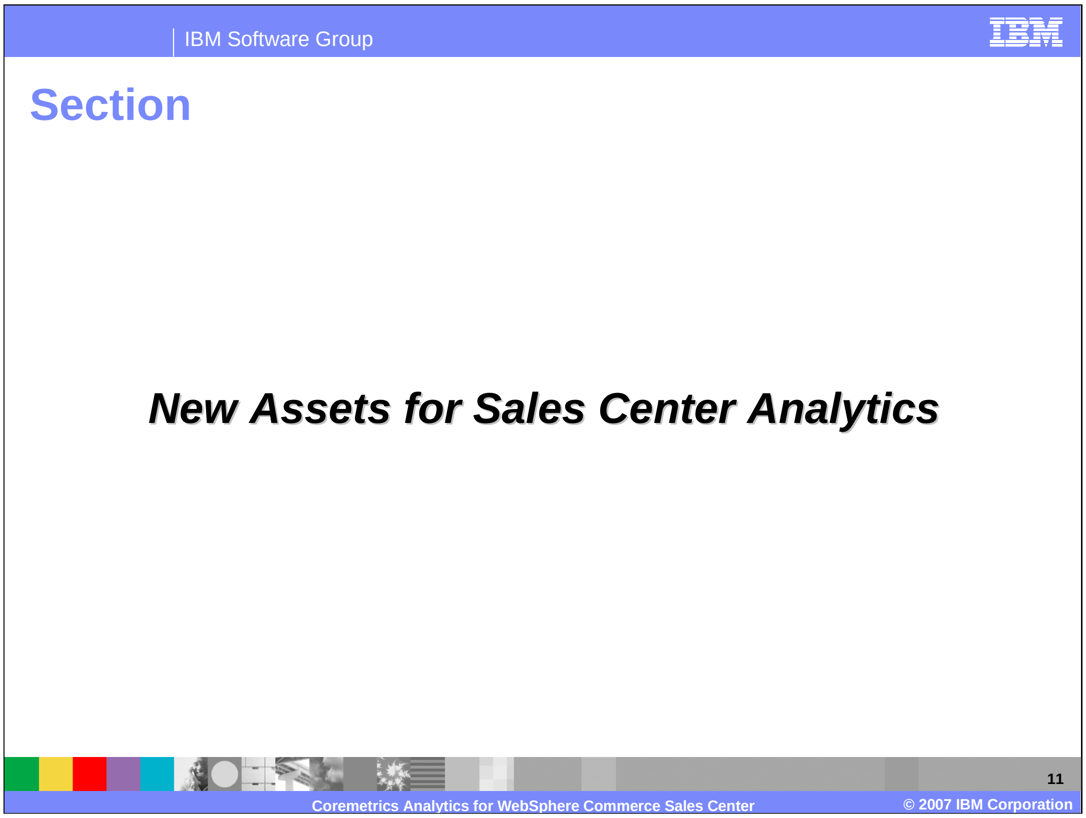
The event data as attachment

The response sent by Coremetrics will consist of the following main items:

A return code of the transaction to indicate success or failure. This is to be defined by Coremetrics.

A recommended action to take such as retry, abort and so on. This is to be defined by Coremetrics.

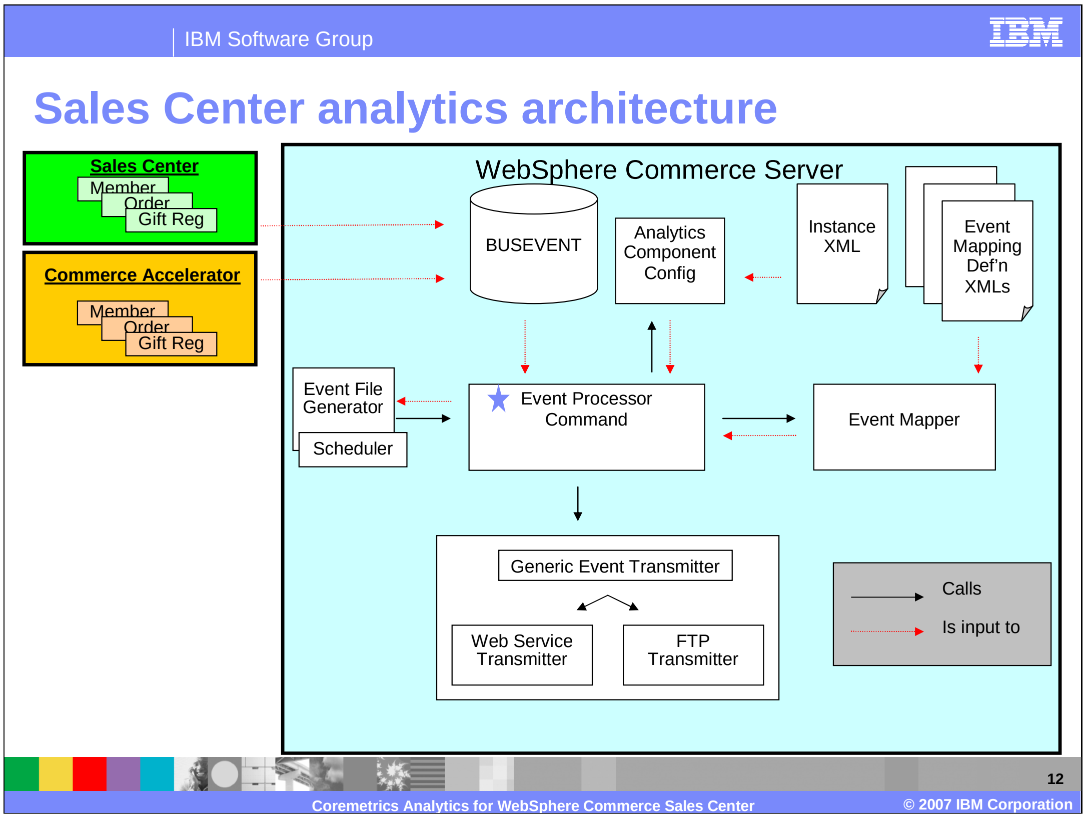
Number of events received and process. This will be used to compare with the number of events actually sent as an additional check.



Section

New Assets for Sales Center Analytics

This section discusses all of the new assets that comprise the Sales Center Analytics flow.



IBM Software Group

IBM

Sales Center analytics architecture

Sales Center

Member

Order

Gift Reg

Commerce Accelerator

Member

Order

Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

Is input to

12

Coremetrics Analytics for WebSphere Commerce Sales Center

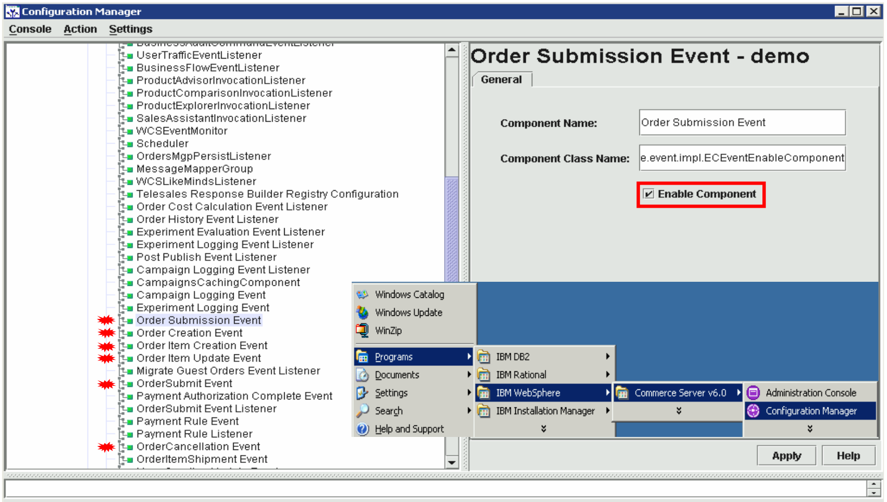
© 2007 IBM Corporation

At a high level, the overall solutions works as follows. Components from the Sales Center and WebSphere Commerce Accelerator raise events which are managed and persisted in the BUSEVENT table by the Event Infrastructure. During enablement of this analytics event feature, a new scheduled task - EventProcessorCmd - is created. This command is invoked periodically by the scheduler. When invoked, it accesses its configuration in the instance XML file through the AnalyticsComponentConfiguration helper class. If there are qualifying (enabled and unprocessed) events in the BUSEVENT table, it will proceed to call the EventMapper helper to help transform the internal Common Base Event (CBE) XML format into an external, consumable format for Coremetrics. The new XML's are then passed along to the appropriate transmitter for the actual communication and transfer of the data.

The EventProcessorCmd can also be invoked by the EventFileGenerator utility. When invoked in this mode, the processor still calls the EventMapper to transform the event XML's, but the output is returned as part of the HTML response, rather than allowing a transmitter implementation to handle the output.

Configuration Manager changes

- Components enabled after feature is enabled



Configuration Manager

Console Action Settings

- UserTrafficEventListener
- BusinessFlowEventListener
- ProductAdvisorInvocationListener
- ProductComparisonInvocationListener
- ProductExplorerInvocationListener
- SalesAssistantInvocationListener
- WCSEventMonitor
- Scheduler
- OrdersMgpPersistListener
- MessageMapperGroup
- WCSLikeMindsListener
- Telesales Response Builder Registry Configuration
- Order Cost Calculation Event Listener
- Order History Event Listener
- Experiment Evaluation Event Listener
- Experiment Logging Event Listener
- Post Publish Event Listener
- Campaign Logging Event Listener
- CampaignsCachingComponent
- Campaign Logging Event
- Experiment Logging Event
- Order Submission Event
- Order Creation Event
- Order Item Creation Event
- Order Item Update Event
- Migrate Guest Orders Event Listener
- OrderSubmit Event
- Payment Authorization Complete Event
- OrderSubmit Event Listener
- Payment Rule Event
- Payment Rule Listener
- OrderCancellation Event
- OrderItemShipment Event

Order Submission Event - demo

General

Component Name: Order Submission Event

Component Class Name: e.event.impl.ECEventEnableComponent

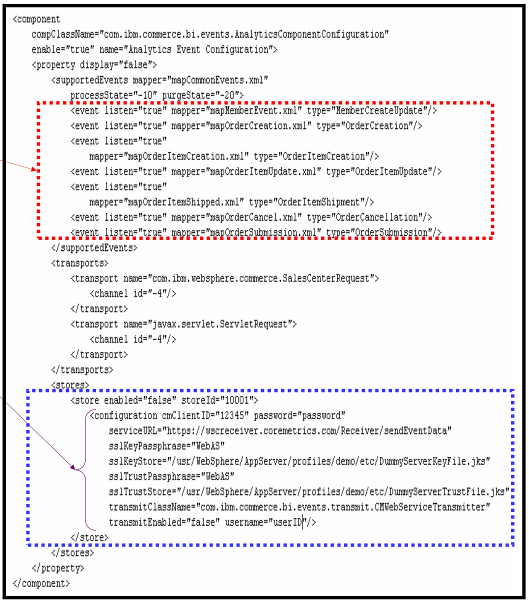
☑ Enable Component

Apply Help

As a result of the feature being enabled, the Configuration Manager has changed. Notice the new events in red that have been added to the Configuration Manager. Here is a screen capture of an Order Submission Event. Note that this event has its check box enabled by default after the feature has been enabled. The Configuration Manager can be started by selecting Start→ Programs→ IBM WebSphere→ Commerce Server V6.0→ Configuration Manager.

Instance XML (wc-server.xml)

- Events listed in supported events tag
- Each store has its own configuration
- Each store has ten mandatory parameters

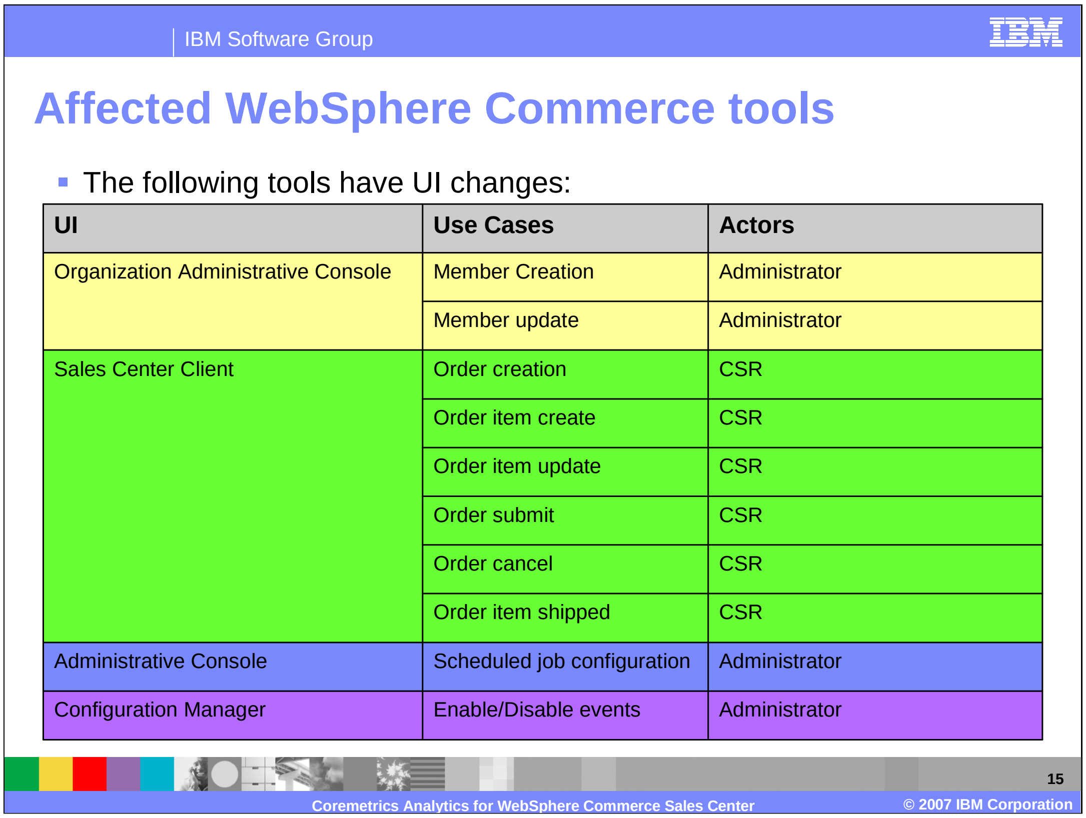


```
<component
   compClassName="com.ibm.commerce.bi.events.AnalyticsComponentConfiguration"
   enable="true" name="Analytics Event Configuration">
   <property display="false">
      <supportedEvents mapper="mapCommonEvents.xml"
         processState="-10" purgeState="-20">
         <event listen="true" mapper="mapMemberEvent.xml" type="MemberCreateUpdate"/>
         <event listen="true" mapper="mapOrderCreation.xml" type="OrderCreation"/>
         <event listen="true"
            mapper="mapOrderItemCreation.xml" type="OrderItemCreation"/>
         <event listen="true" mapper="mapOrderItemUpdate.xml" type="OrderItemUpdate"/>
         <event listen="true"
            mapper="mapOrderItemShipped.xml" type="OrderItemShipment"/>
         <event listen="true" mapper="mapOrderCancel.xml" type="OrderCancellation"/>
         <event listen="true" mapper="mapOrderSubmission.xml" type="OrderSubmission"/>
      </supportedEvents>
      <transports>
         <transport name="com.ibm.websphere.commerce.SalesCenterRequest">
            <channel id="-4"/>
         </transport>
         <transport name="javax.servlet.ServletRequest">
            <channel id="-4"/>
         </transport>
      </transports>
      <stores>
         <store enabled="false" storeId="10001">
            <configuration cmClientID="12345" password="password"
               serviceURL="https://wscreceiver.coremetrics.com/Receiver/sendEventData"
               sslKeyPassphrase="WebAS"
               sslKeyStore="/usr/WebSphere/AppServer/profiles/demo/etc/DummyServerKeyFile.jks"
               sslTrustPassphrase="WebAS"
               sslTrustStore="/usr/WebSphere/AppServer/profiles/demo/etc/DummyServerTrustFile.jks"
               transmitClassName="com.ibm.commerce.bi.events.transmit.CMWebServiceTransmitter"
               transmitEnabled="false" username="userID"/>
         </store>
      </stores>
   </property>
</component>
```

The *supportedEvents* block lists the various events that could be generated by the WebSphere Commerce Event Infrastructure with each having a *listen* flag. This allows the customization of which events to process.

Each store will have its own configuration. The *enabled* attribute tells the EventProcessor which store events to watch for and process. The *configuration* child node has a series of attributes in which only two are mandatory – *transmitEnabled* and *transmitClassName*. All other attributes will be passed to the class defined in *transmitClassName* for its use.

For the Coremetrics Web service implementation, additional parameters such as client ID, username, password and service URL are required. The username and password is the same information as provided to Coremetrics if you have taken advantage of the Closed Loop Segmentation feature. The username must be a WebSphere Commerce user with proper Marketing Manager roles for the store and the encrypted password string is obtained by encrypting a user's plaintext password with the wcs_encrypt utility.



IBM Software Group

Affected WebSphere Commerce tools

- The following tools have UI changes:

UI	Use Cases	Actors
Organization Administrative Console	Member Creation	Administrator
	Member update	Administrator
Sales Center Client	Order creation	CSR
	Order item create	CSR
	Order item update	CSR
	Order submit	CSR
	Order cancel	CSR
	Order item shipped	CSR
Administrative Console	Scheduled job configuration	Administrator
Configuration Manager	Enable/Disable events	Administrator

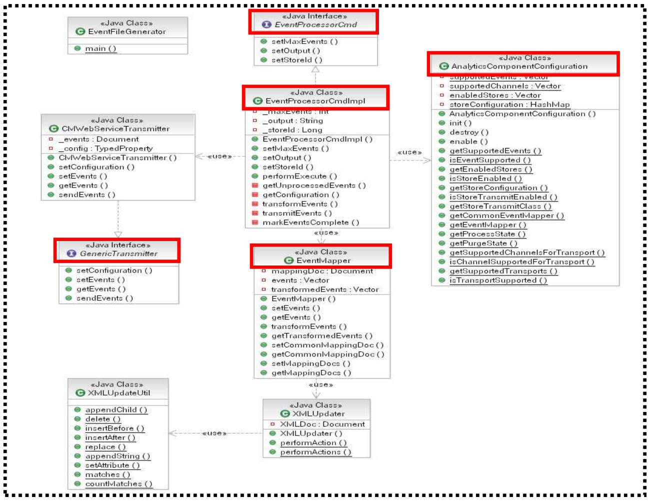
15

Coremetrics Analytics for WebSphere Commerce Sales Center

Here you can see the WebSphere Commerce tools, use cases and Actors affected when this feature is enabled.

Sales Center analytics model

- Classes denoted by red box will be exposed for generic use.



«Java Class»
EventFileGenerator
main ()

«Java Interface»
EventProcessorCmd
setMaxEvents ()
setOutput ()
setStoreId ()

«Java Class»
AnalyticsComponentConfiguration
supportedEvents : Vector
supportedChannels : Vector
enabledStores : Vector
storeConfiguration : HashMap
AnalyticsComponentConfiguration ()
init ()
destroy ()
enable ()
getSupportedEvents ()
isEventSupported ()
getEnabledStores ()
isStoreEnabled ()
getStoreConfiguration ()
isStoreTransmitEnabled ()
getStoreTransmitClass ()
getCommonEventMapper ()
getEventMapper ()
getProcessState ()
getPurgeState ()
getSupportedChannelsForTransport ()
isChannelSupportedForTransport ()
getSupportedTransports ()
isTransportSupported ()

«Java Class»
EventProcessorCmdImpl
_maxEvents : int
_output : String
_storeId : Long
EventProcessorCmdImpl ()
setMaxEvents ()
setOutput ()
setStoreId ()
performExecute ()
getUnprocessedEvents ()
getConfiguration ()
transformEvents ()
transmitEvents ()
markEventsComplete ()

«Java Class»
CMWebServiceTransmitter
_events : Document
_config : TypedProperty
CMWebServiceTransmitter ()
setConfiguration ()
setEvents ()
getEvents ()
sendEvents ()

«use»

«use»

«use»

«Java Interface»
GenericTransmitter
setConfiguration ()
setEvents ()
getEvents ()
sendEvents ()

«Java Class»
EventMapper
mappingDoc : Document
events : Vector
transformedEvents : Vector
EventMapper ()
setEvents ()
getEvents ()
transformEvents ()
getTransformedEvents ()
setCommonMappingDoc ()
getCommonMappingDoc ()
setMappingDocs ()
getMappingDocs ()

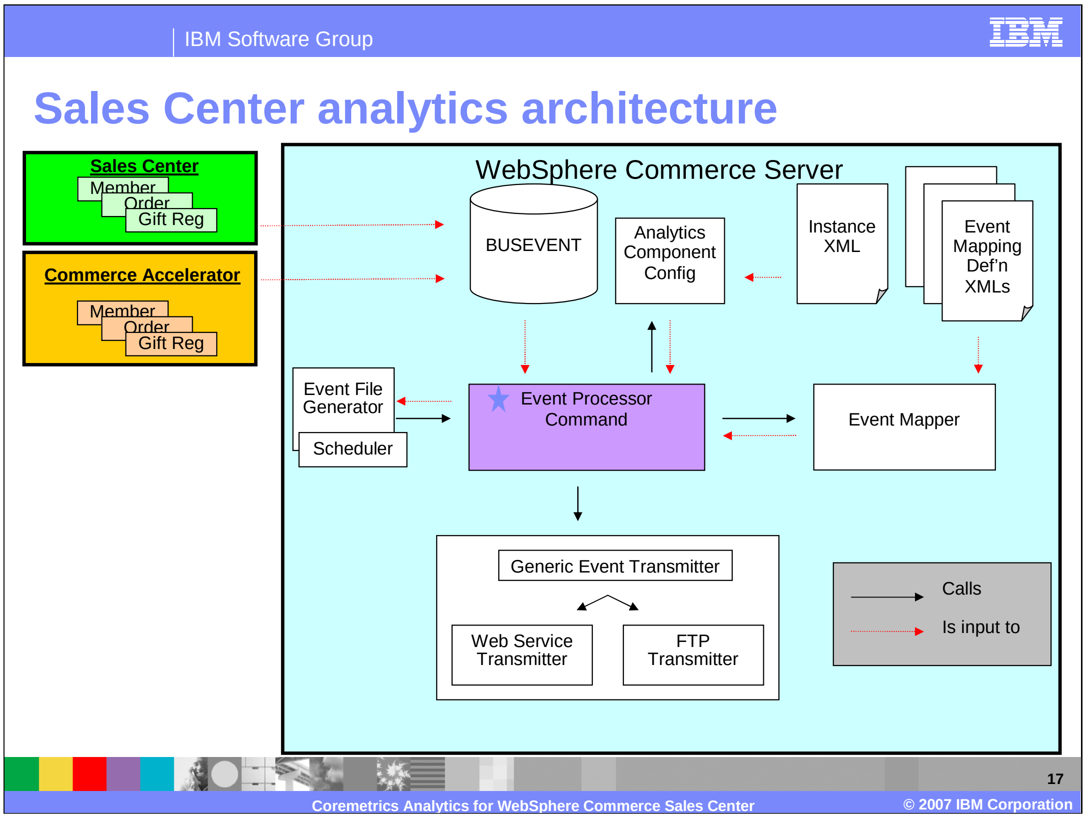
«use»

«Java Class»
XMLUpdateUtil
appendChild ()
delete ()
insertBefore ()
insertAfter ()
replace ()
appendString ()
setAttribute ()
matches ()
countMatches ()

«use»

«Java Class»
XMLUpdater
XMLDoc : Document
XMLUpdater ()
performAction ()
performActions ()

This class hierarchy diagram supports the Sales Center Analytics flow that was shown a few slides ago. The classes denoted by the red box will have their JavaDoc and API exposed for generic use and customization purposes.



IBM Software Group

Sales Center analytics architecture

Sales Center
Member
Order
Gift Reg

Commerce Accelerator
Member
Order
Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

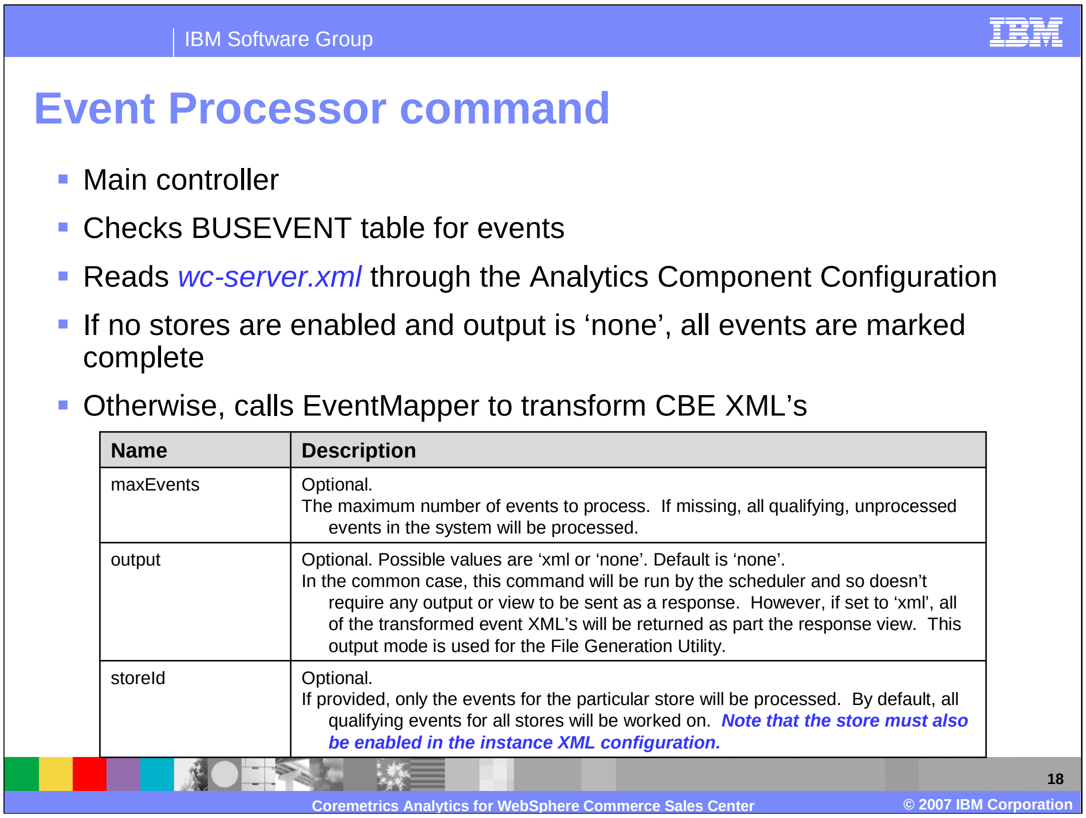
Is input to

17

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

This portion of the presentation will discuss each particular piece in this flow starting with the Event Processor Command.



IBM Software Group

Event Processor command

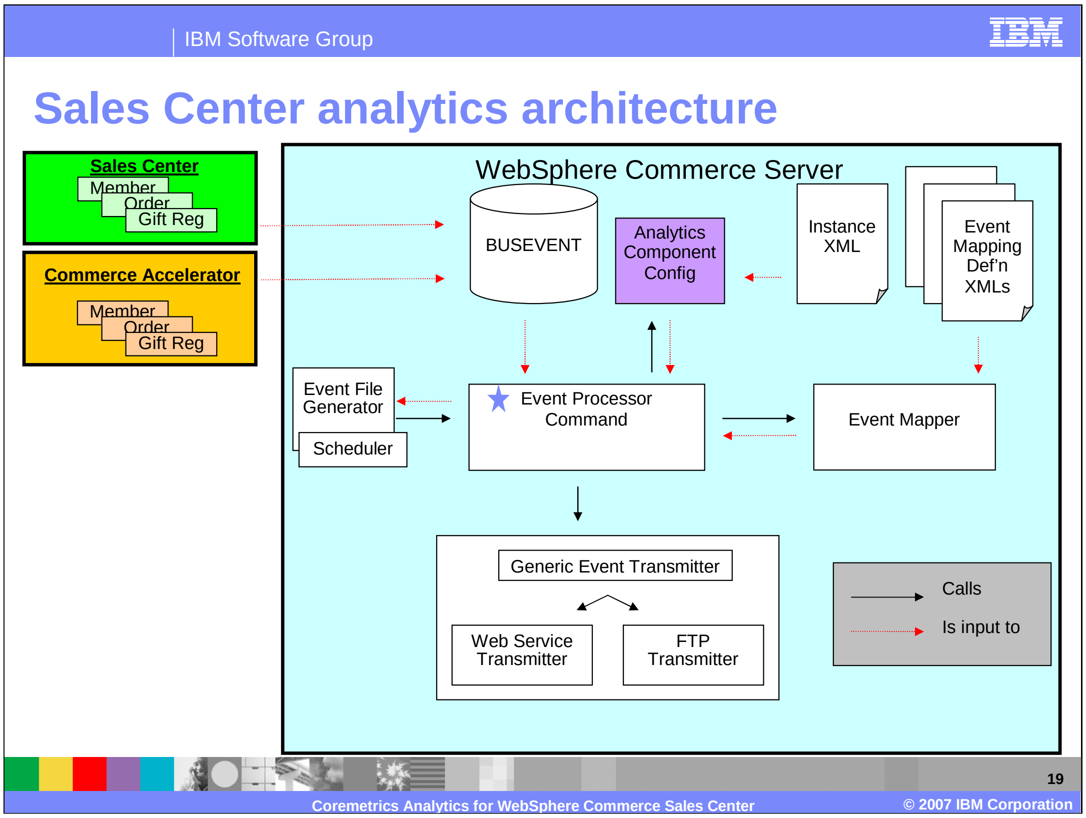
- Main controller
- Checks BUSEVENT table for events
- Reads *wc-server.xml* through the Analytics Component Configuration
- If no stores are enabled and output is ‘none’, all events are marked complete
- Otherwise, calls EventMapper to transform CBE XML’s

Name	Description
maxEvents	Optional. The maximum number of events to process. If missing, all qualifying, unprocessed events in the system will be processed.
output	Optional. Possible values are ‘xml or ‘none’. Default is ‘none’. In the common case, this command will be run by the scheduler and so doesn’t require any output or view to be sent as a response. However, if set to ‘xml’, all of the transformed event XML’s will be returned as part the response view. This output mode is used for the File Generation Utility.
storeId	Optional. If provided, only the events for the particular store will be processed. By default, all qualifying events for all stores will be worked on. ***Note that the store must also be enabled in the instance XML configuration.***

18

Coremetrics Analytics for WebSphere Commerce Sales Center

This command is the main controller that handles the Analytics Event component and will only be made available through the Site Administration Web module. It first checks the BUSEVENT table for unprocessed events. Configuration of this component is persisted in the instance XML (wc-server.xml) and accessed through the *AnalyticsComponentConfiguration*. If no stores have a transmitter enabled and the *output* parameter is ‘none’, it will immediately mark all of the events as complete and quit so that the WebSphere Commerce Event Infrastructure can continue to manage the event system. Otherwise, the EventMapper is called in order to transform the internal CBE XML’s into an external, consumable format. The transformed XML’s are then sent as the response if *output* is set to ‘xml’, or they are passed along to an implementation of GenericTransmitter for some external system.



IBM Software Group

Sales Center analytics architecture

Sales Center
Member
Order
Gift Reg

Commerce Accelerator
Member
Order
Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

Is input to

19

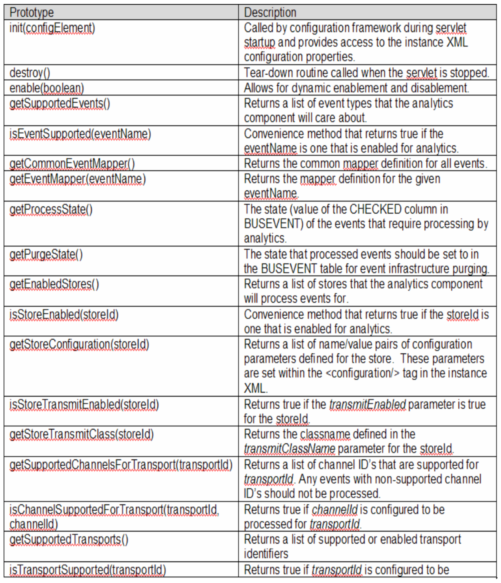
Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Analytics Component Configuration will now be discussed.

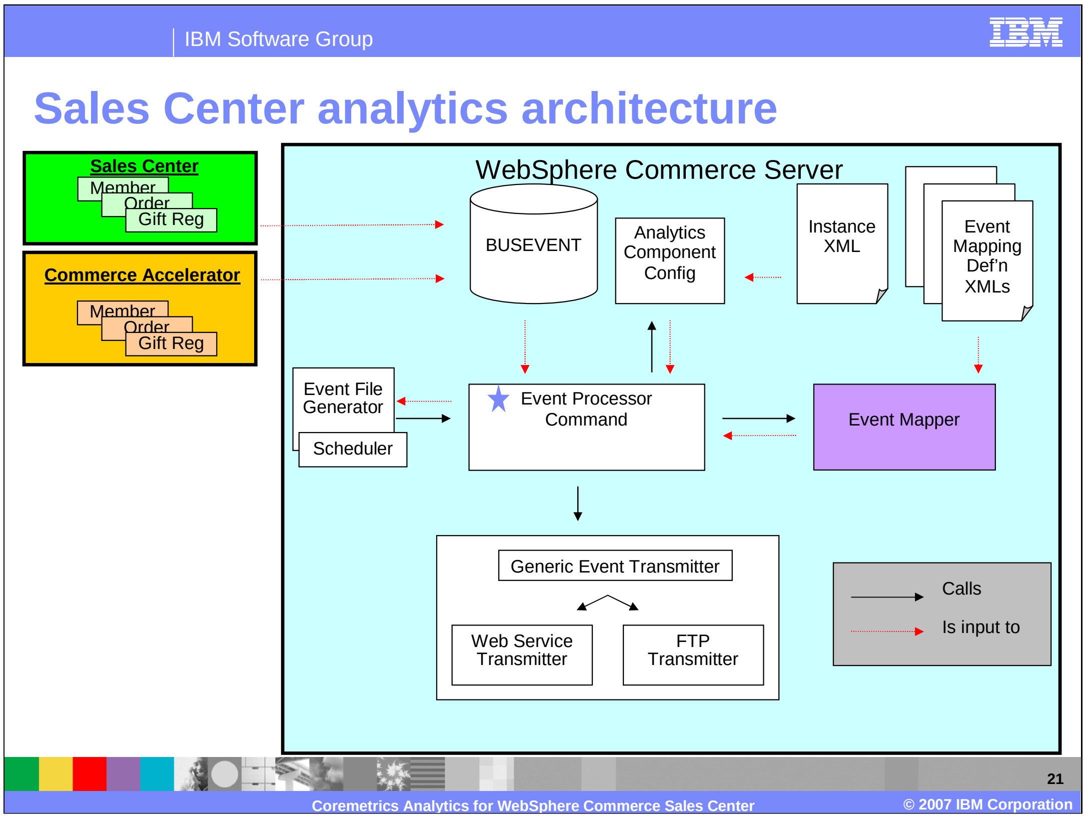
Analytics Component Configuration

- Provides access to configuration parameters in wc-server.xml



Prototype	Description
init(configElement)	Called by configuration framework during servlet startup and provides access to the instance XML configuration properties.
destroy()	Tear-down routine called when the servlet is stopped.
enable(boolean)	Allows for dynamic enablement and disablement.
getSupportedEvents()	Returns a list of event types that the analytics component will care about.
isEventSupported(eventName)	Convenience method that returns true if the eventName is one that is enabled for analytics.
getCommonEventMapper()	Returns the common mapper definition for all events.
getEventMapper(eventName)	Returns the mapper definition for the given eventName.
getProcessState()	The state (value of the CHECKED column in BUSEVENT) of the events that require processing by analytics.
getPurgeState()	The state that processed events should be set to in the BUSEVENT table for event infrastructure purging.
getEnabledStores()	Returns a list of stores that the analytics component will process events for.
isStoreEnabled(storeId)	Convenience method that returns true if the storeId is one that is enabled for analytics.
getStoreConfiguration(storeId)	Returns a list of name/value pairs of configuration parameters defined for the store. These parameters are set within the <configuration/> tag in the instance XML.
isStoreTransmitEnabled(storeId)	Returns true if the *transmitEnabled* parameter is true for the storeId.
getStoreTransmitClass(storeId)	Returns the classname defined in the *transmitClassName* parameter for the storeId.
getSupportedChannelsForTransport(transportId)	Returns a list of channel ID's that are supported for *transportId*. Any events with non-supported channel ID's should not be processed.
isChannelSupportedForTransport(transportId, channelId)	Returns true if *channelId* is configured to be processed for *transportId*.
getSupportedTransports()	Returns a list of supported or enabled transport identifiers
isTransportSupported(transportId)	Returns true if *transportId* is configured to be

This helper class provides static access to the configuration parameters defined in the instance XML.



IBM Software Group

Sales Center analytics architecture

Sales Center
Member
Order
Gift Reg

Commerce Accelerator
Member
Order
Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

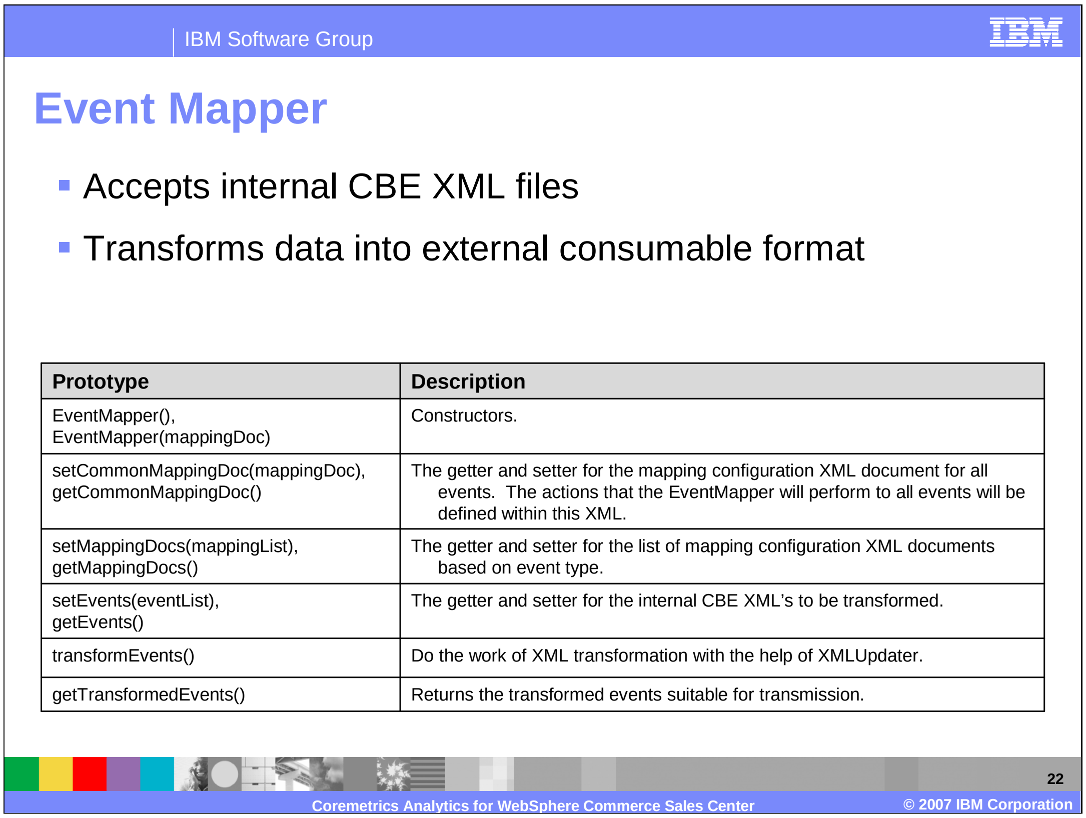
→	Calls
⇢	Is input to

21

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Event Mapper will now be discussed.

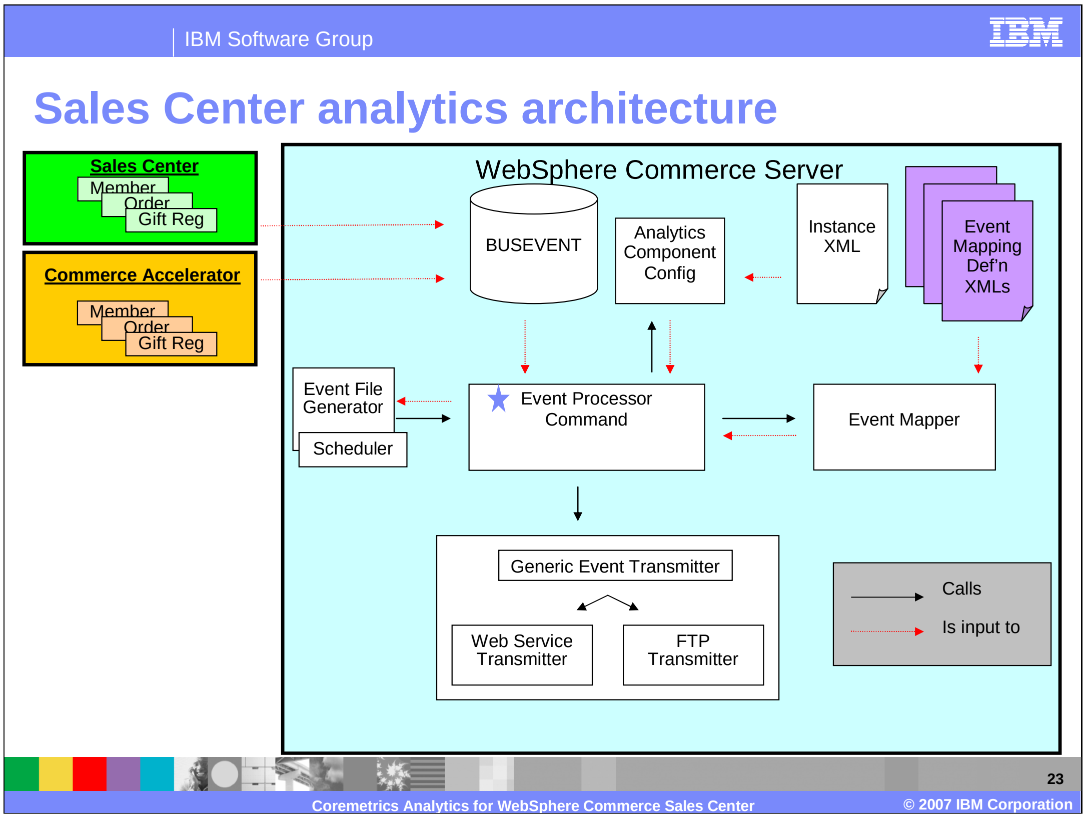


Event Mapper

- Accepts internal CBE XML files
- Transforms data into external consumable format

Prototype	Description
EventMapper(), EventMapper(mappingDoc)	Constructors.
setCommonMappingDoc(mappingDoc), getCommonMappingDoc()	The getter and setter for the mapping configuration XML document for all events. The actions that the EventMapper will perform to all events will be defined within this XML.
setMappingDocs(mappingList), getMappingDocs()	The getter and setter for the list of mapping configuration XML documents based on event type.
setEvents(eventList), getEvents()	The getter and setter for the internal CBE XML's to be transformed.
transformEvents()	Do the work of XML transformation with the help of XMLUpdater.
getTransformedEvents()	Returns the transformed events suitable for transmission.

Given a list of internal CBE XML files, the Event Mapper transforms them into an external format so they can be consumed by non-WebSphere Commerce systems. For example, internal formats mostly contain ID values of things like products and orders. Specific details can be found by a simple database query within WebSphere Commerce. The problem with external systems, is that they don't have access to the database. As a result, this transformation process does the data lookup and sends them along. Also, other types of transformation such as data addition, removal or modification are supported here.



IBM Software Group

Sales Center analytics architecture

Sales Center

Member

Order

Gift Reg

Commerce Accelerator

Member

Order

Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

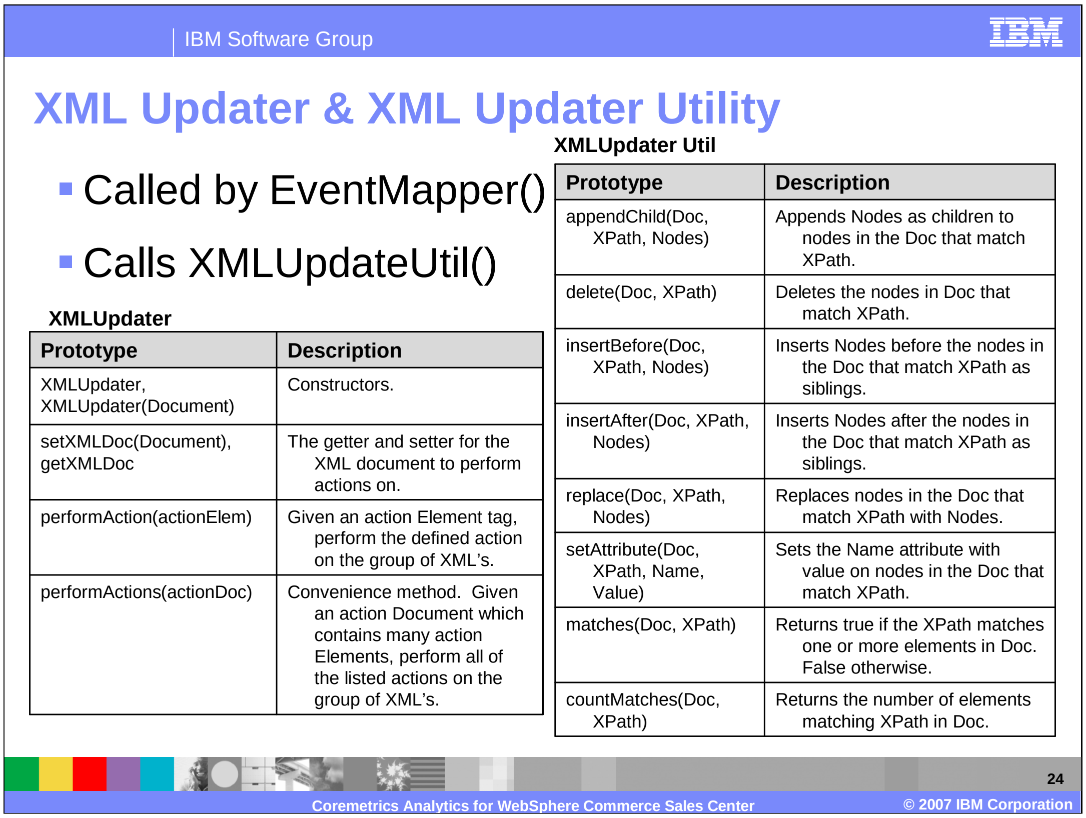
Is input to

23

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Event Mapping Definition XML’s will now be discussed.



XML Updater & XML Updater Utility

- Called by EventMapper()
- Calls XMLUpdateUtil()

XMLUpdater

Prototype	Description
XMLUpdater, XMLUpdater(Document)	Constructors.
setXMLDoc(Document), getXMLDoc	The getter and setter for the XML document to perform actions on.
performAction(actionElem)	Given an action Element tag, perform the defined action on the group of XML's.
performActions(actionDoc)	Convenience method. Given an action Document which contains many action Elements, perform all of the listed actions on the group of XML's.

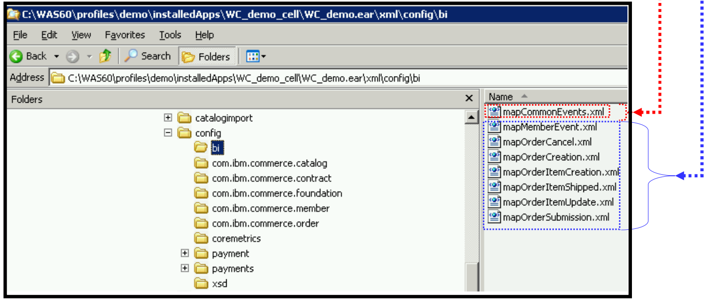
XMLUpdater Util

Prototype	Description
appendChild(Doc, XPath, Nodes)	Appends Nodes as children to nodes in the Doc that match XPath.
delete(Doc, XPath)	Deletes the nodes in Doc that match XPath.
insertBefore(Doc, XPath, Nodes)	Inserts Nodes before the nodes in the Doc that match XPath as siblings.
insertAfter(Doc, XPath, Nodes)	Inserts Nodes after the nodes in the Doc that match XPath as siblings.
replace(Doc, XPath, Nodes)	Replaces nodes in the Doc that match XPath with Nodes.
setAttribute(Doc, XPath, Name, Value)	Sets the Name attribute with value on nodes in the Doc that match XPath.
matches(Doc, XPath)	Returns true if the XPath matches one or more elements in Doc. False otherwise.
countMatches(Doc, XPath)	Returns the number of elements matching XPath in Doc.

These two public classes are utility classes that contain convenience static methods that manipulate XML documents. These classes do all of the work of appending, deleting, replacing and adding data to the XML file.

XML Mapping document definitions

- Used by EventMapper to transform CBE XML's
- Located in <EAR>/xml/config/bi
- Must adhere to DTD
- Utilize mapCommonEvents.xml
- Each event type has its own mapper



C:\WAS60\profiles\demo\installedApps\WC_demo_cell\WC_demo.ear\xml\config\bi

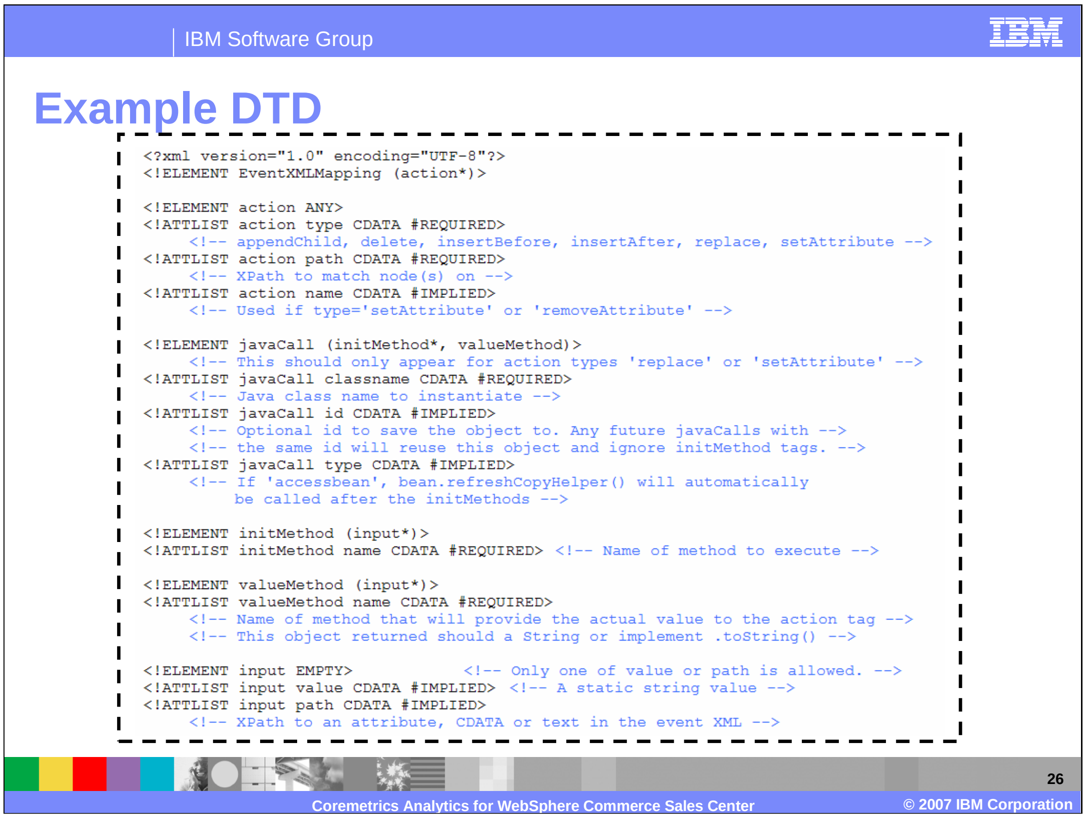
Address C:\WAS60\profiles\demo\installedApps\WC_demo_cell\WC_demo.ear\xml\config\bi

- catalogimport
- config
  - bi
  - com.ibm.commerce.catalog
  - com.ibm.commerce.contract
  - com.ibm.commerce.foundation
  - com.ibm.commerce.member
  - com.ibm.commerce.order
  - coremetrics
  - payment
  - payments
  - xsd

Name

- mapCommonEvents.xml
- mapMemberEvent.xml
- mapOrderCancel.xml
- mapOrderCreation.xml
- mapOrderItemCreation.xml
- mapOrderItemShipped.xml
- mapOrderItemUpdate.xml
- mapOrderSubmission.xml

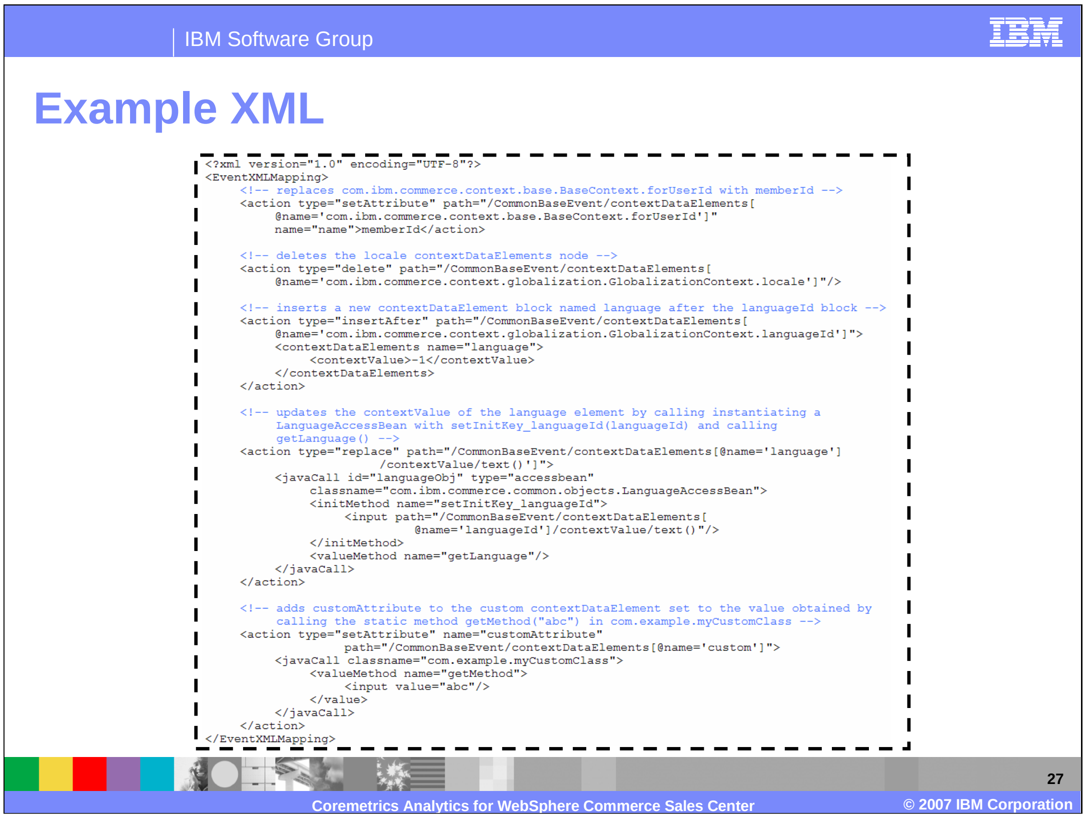
The event mapper definition files are used by *EventMapper* to transform CBE XML's. These files are located in the <EAR>/xml/config/bi directory and they must adhere to the document type definition (DTD) shown on the next slide. A common mapper definition file will be defined in the mapCommonEvents.xml file. The mapping defined within this file will be applied to all CBE XML documents and is useful for handling common XML nodes such as the *contextDataElements* tags. Each event type will then have their own mapper for event-specific transformations. These are all defined in the instance XML configuration.



Example DTD

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT EventXMLMapping (action*)>

<!ELEMENT action ANY>
<!ATTLIST action type CDATA #REQUIRED>
     <!-- appendChild, delete, insertBefore, insertAfter, replace, setAttribute -->
<!ATTLIST action path CDATA #REQUIRED>
     <!-- XPath to match node(s) on -->
<!ATTLIST action name CDATA #IMPLIED>
     <!-- Used if type='setAttribute' or 'removeAttribute' -->

<!ELEMENT javaCall (initMethod*, valueMethod)>
     <!-- This should only appear for action types 'replace' or 'setAttribute' -->
<!ATTLIST javaCall classname CDATA #REQUIRED>
     <!-- Java class name to instantiate -->
<!ATTLIST javaCall id CDATA #IMPLIED>
     <!-- Optional id to save the object to. Any future javaCalls with -->
     <!-- the same id will reuse this object and ignore initMethod tags. -->
<!ATTLIST javaCall type CDATA #IMPLIED>
     <!-- If 'accessbean', bean.refreshCopyHelper() will automatically
          be called after the initMethods -->

<!ELEMENT initMethod (input*)>
<!ATTLIST initMethod name CDATA #REQUIRED> <!-- Name of method to execute -->

<!ELEMENT valueMethod (input*)>
<!ATTLIST valueMethod name CDATA #REQUIRED>
     <!-- Name of method that will provide the actual value to the action tag -->
     <!-- This object returned should a String or implement .toString() -->

<!ELEMENT input EMPTY>            <!-- Only one of value or path is allowed. -->
<!ATTLIST input value CDATA #IMPLIED> <!-- A static string value -->
<!ATTLIST input path CDATA #IMPLIED>
     <!-- XPath to an attribute, CDATA or text in the event XML -->
```

The event mapper XML files must adhere to the rules specified in this DTD.



IBM Software Group

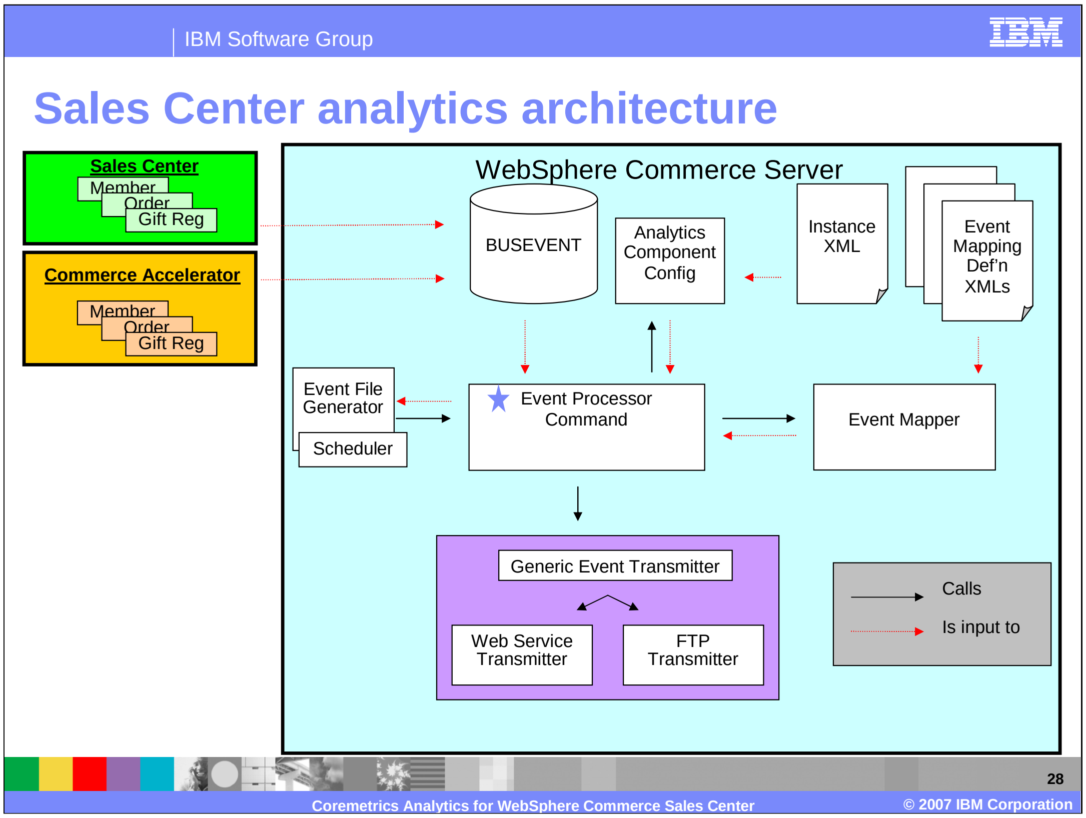
Example XML

```
<?xml version="1.0" encoding="UTF-8"?>
<EventXMLMapping>
     <!-- replaces com.ibm.commerce.context.base.BaseContext.forUserId with memberId -->
     <action type="setAttribute" path="/CommonBaseEvent/contextDataElements[
          @name='com.ibm.commerce.context.base.BaseContext.forUserId']"
          name="name">memberId</action>

     <!-- deletes the locale contextDataElements node -->
     <action type="delete" path="/CommonBaseEvent/contextDataElements[
          @name='com.ibm.commerce.context.globalization.GlobalizationContext.locale']"/>

     <!-- inserts a new contextDataElement block named language after the languageId block -->
     <action type="insertAfter" path="/CommonBaseEvent/contextDataElements[
          @name='com.ibm.commerce.context.globalization.GlobalizationContext.languageId']">
          <contextDataElements name="language">
               <contextValue>-1</contextValue>
          </contextDataElements>
     </action>

     <!-- updates the contextValue of the language element by calling instantiating a
          LanguageAccessBean with setInitKey_languageId(languageId) and calling
          getLanguage() -->
     <action type="replace" path="/CommonBaseEvent/contextDataElements[@name='language']
                         /contextValue/text()']">
          <javaCall id="languageObj" type="accessbean"
               classname="com.ibm.commerce.common.objects.LanguageAccessBean">
               <initMethod name="setInitKey_languageId">
                    <input path="/CommonBaseEvent/contextDataElements[
                              @name='languageId']/contextValue/text()"/>
               </initMethod>
               <valueMethod name="getLanguage"/>
          </javaCall>
     </action>

     <!-- adds customAttribute to the custom contextDataElement set to the value obtained by
          calling the static method getMethod("abc") in com.example.myCustomClass -->
     <action type="setAttribute" name="customAttribute"
                    path="/CommonBaseEvent/contextDataElements[@name='custom']">
          <javaCall classname="com.example.myCustomClass">
               <valueMethod name="getMethod">
                    <input value="abc"/>
               </value>
          </javaCall>
     </action>
</EventXMLMapping>
```

27

Coremetrics Analytics for WebSphere Commerce Sales Center

As you can see within the mapCommonEvents.xml file, there are many things happening to each event as it is transformed from internal to external format. Yet, this file along with the DTD can be customized to fit your needs.



IBM Software Group

IBM

Sales Center analytics architecture

Sales Center

Member

Order

Gift Reg

Commerce Accelerator

Member

Order

Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

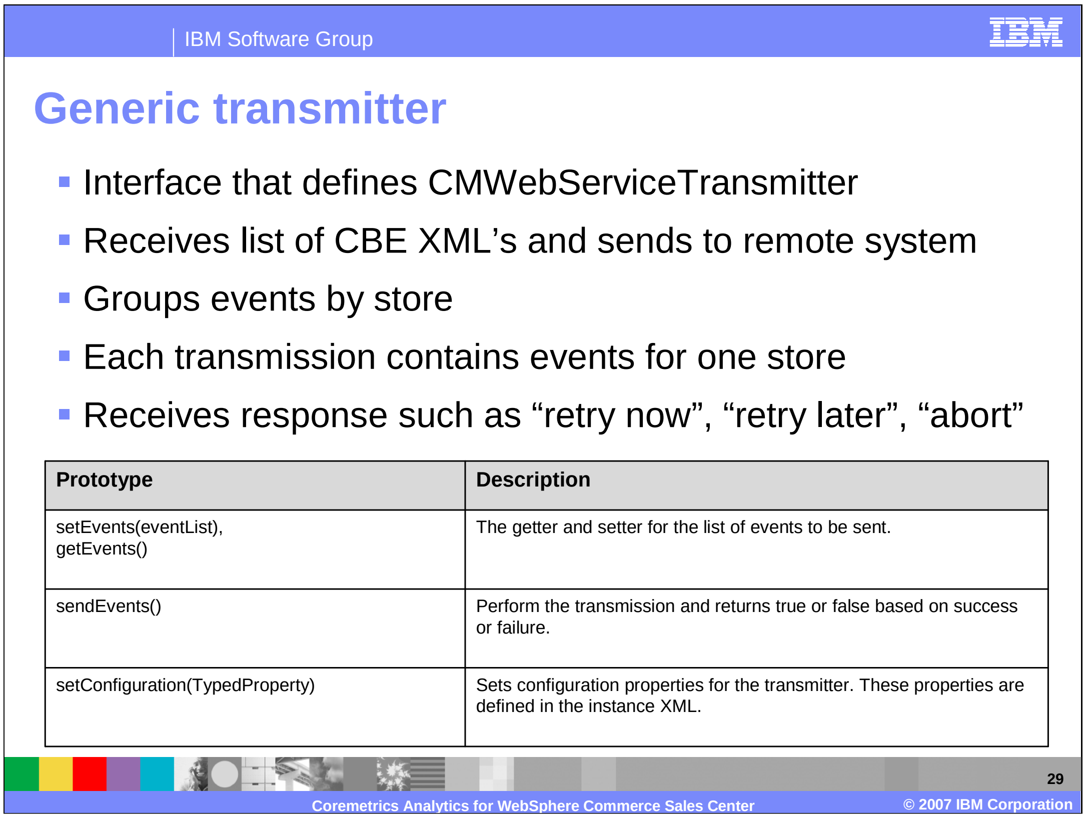
Is input to

28

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Generic Event Transmitter will now be discussed.

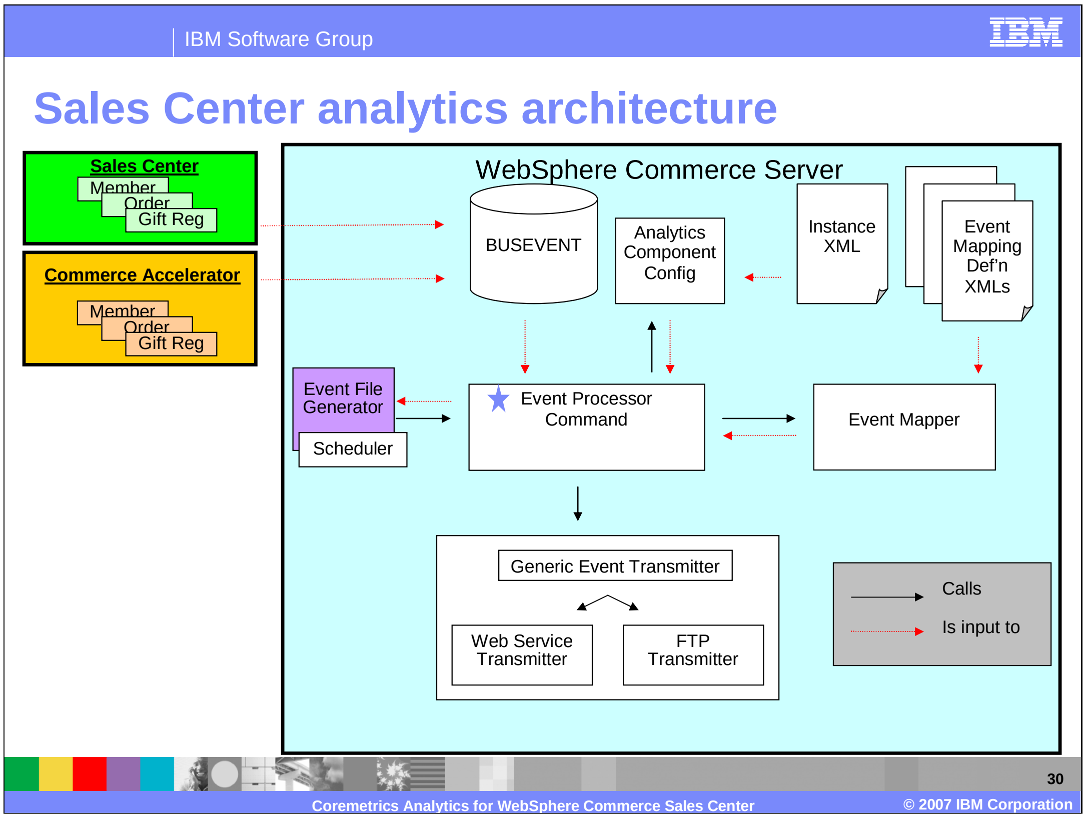


Generic transmitter

- Interface that defines CMWebServiceTransmitter
- Receives list of CBE XML’s and sends to remote system
- Groups events by store
- Each transmission contains events for one store
- Receives response such as “retry now”, “retry later”, “abort”

Prototype	Description
setEvents(eventList), getEvents()	The getter and setter for the list of events to be sent.
sendEvents()	Perform the transmission and returns true or false based on success or failure.
setConfiguration(TypedProperty)	Sets configuration properties for the transmitter. These properties are defined in the instance XML.

The GenericTransmitter is an interface that defines a transmitter that will receive a list of CBE XML’s and sends it to some remote system. Currently, only CMWebServiceTransmitter is implemented – other transmitters like FTP and e-mail, may be subsequently implemented by a customer. Transmissions performed by CMWebServiceTransmitter will be grouped by store. If more than one store is enabled in the configuration, each request made to Coremetrics will contain events for one store. As part of the response, Coremetrics may also supply a recommended action. These have yet to be defined by Coremetrics, but may include things such as “retry now”, “retry later”, “about”, and so on. Based on the recommendation, action may be taken and handled within this transmitter.



IBM Software Group

IBM

Sales Center analytics architecture

Sales Center

Member

Order

Gift Reg

Commerce Accelerator

Member

Order

Gift Reg

WebSphere Commerce Server

BUSEVENT

Analytics Component Config

Instance XML

Event Mapping Def'n XMLs

Event File Generator

Scheduler

Event Processor Command

Event Mapper

Generic Event Transmitter

Web Service Transmitter

FTP Transmitter

Calls

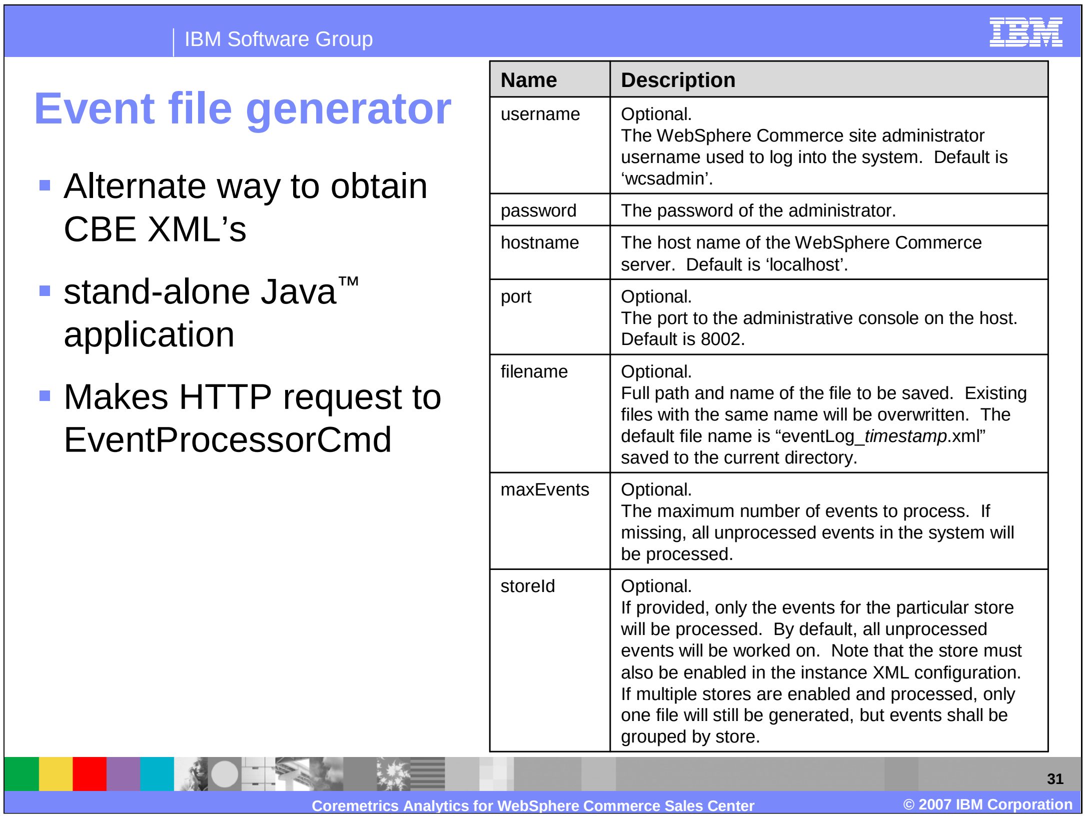
Is input to

30

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Event File Generator will now be discussed.



IBM Software Group

Event file generator

- Alternate way to obtain CBE XML's
- stand-alone Java™ application
- Makes HTTP request to EventProcessorCmd

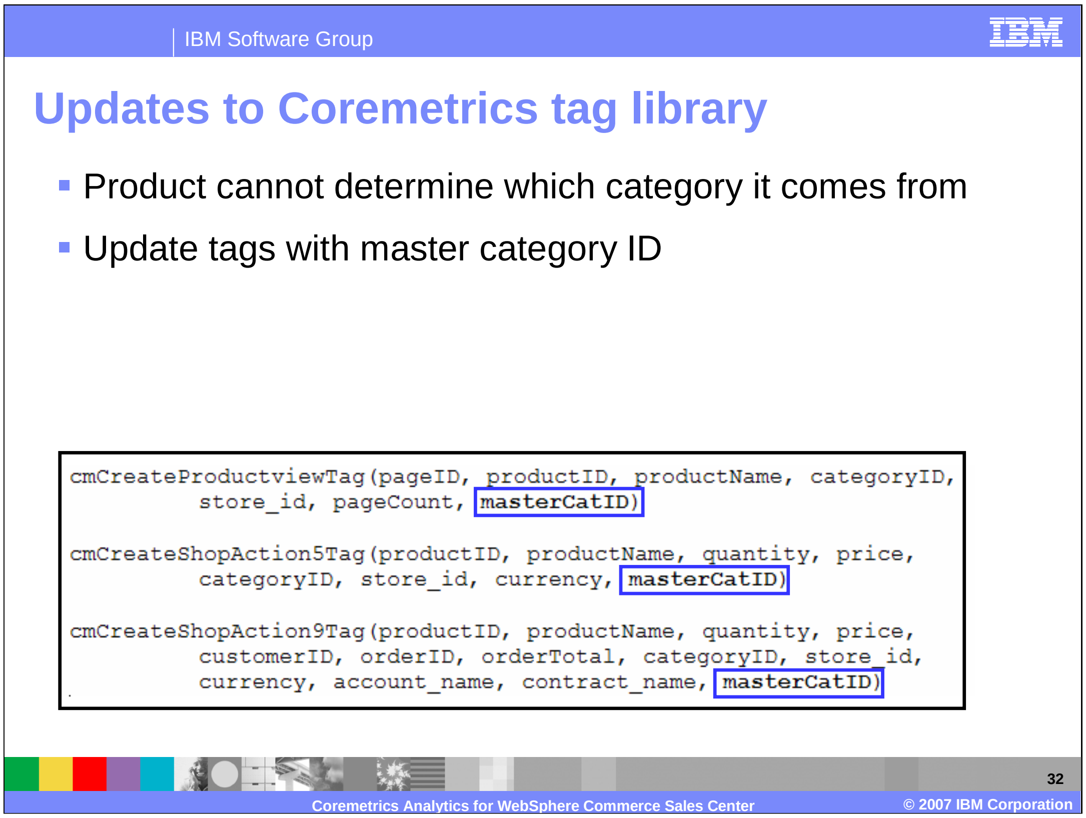
Name	Description
username	Optional. The WebSphere Commerce site administrator username used to log into the system. Default is 'wcsadmin'.
password	The password of the administrator.
hostname	The host name of the WebSphere Commerce server. Default is 'localhost'.
port	Optional. The port to the administrative console on the host. Default is 8002.
filename	Optional. Full path and name of the file to be saved. Existing files with the same name will be overwritten. The default file name is "eventLog_*timestamp*.xml" saved to the current directory.
maxEvents	Optional. The maximum number of events to process. If missing, all unprocessed events in the system will be processed.
storeId	Optional. If provided, only the events for the particular store will be processed. By default, all unprocessed events will be worked on. Note that the store must also be enabled in the instance XML configuration. If multiple stores are enabled and processed, only one file will still be generated, but events shall be grouped by store.

31

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

The Event File Generator is a stand-alone Java application meant to be run on the command line by way of a shell/batch script. The goal is to provide an alternate means to obtain the transformed CBE XML's suitable for external systems if a customer chooses to disable automatic transmission. This is achieved by making a HTTP request to the Commerce server to the *EventProcessorCmd* with the output parameter set to 'xml'. To handle SSL keys and certificates, this application will accept all certificate authorities to simplify the usage by customers.



IBM Software Group

Updates to Coremetrics tag library

- Product cannot determine which category it comes from
- Update tags with master category ID

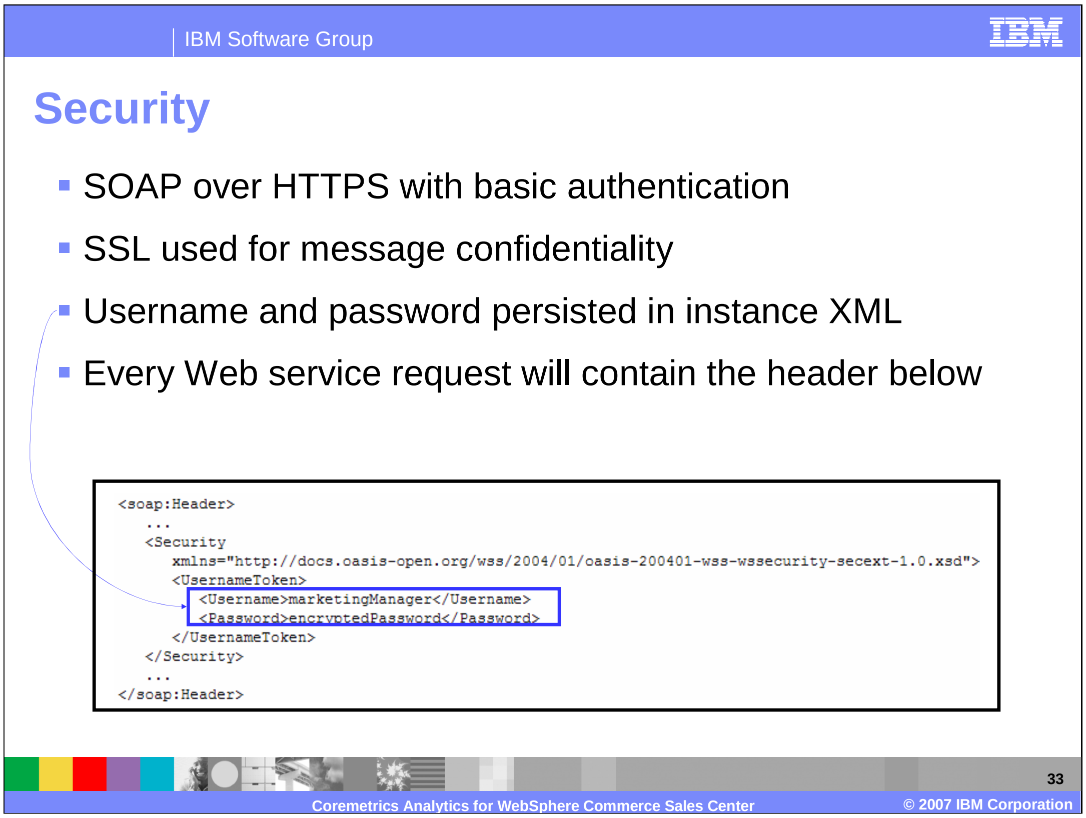
```
cmCreateProductviewTag(pageID, productID, productName, categoryID,
         store_id, pageCount, masterCatID)

cmCreateShopAction5Tag(productID, productName, quantity, price,
         categoryID, store_id, currency, masterCatID)

cmCreateShopAction9Tag(productID, productName, quantity, price,
         customerID, orderID, orderTotal, categoryID, store_id,
         currency, account_name, contract_name, masterCatID)
```

32

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

There is a limitation in the Coremetrics tag library implementation (specifically in the Product tag) in that when a product is shown, it has no idea which category it came from. Consider the case where a sales catalog is used and a product is in multiple categories. For example, a t-shirt under a "Shirts" category is cross-listed in a "Current Sales" category. In this case, the code will choose the category with the smallest ID in the database and associate that category to the product. As a result, the Coremetrics report may never show the t-shirt statistics under the “Current Sales” category. So you update the Product, Cart (Shop Action 5), and Order (Shop Action 9) tags with an addition parameter master category ID to product view tag to identify the master category and to produce the Master Merchandise Report. In case you are wondering, the Shop Action 5 tag captures data about what items were present in a customer’s shopping cart and the Shop Action 9 tag captures data about what products were purchased by a customer



Security

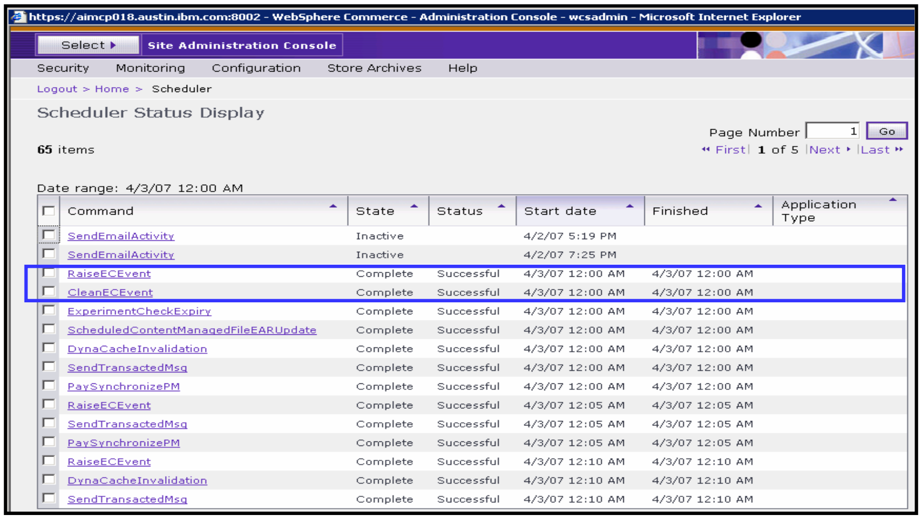
- SOAP over HTTPS with basic authentication
- SSL used for message confidentiality
- Username and password persisted in instance XML
- Every Web service request will contain the header below

```
<soap:Header>
   ...
   <Security
      xmlns="http://docs.oasis-open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <UsernameToken>
         <Username>marketingManager</Username>
         <Password>encryptedPassword</Password>
      </UsernameToken>
   </Security>
   ...
</soap:Header>
```

For security, SOAP over HTTPS will be used with basic authentication. Transport level security (SSL) will be utilized to maintain message confidentiality. The username and password value is persisted in the instance XML, as a result, every Web service request made to Coremetrics will include the header as shown.

Serviceability

- Use Administrative Console to view scheduler status



https://aimcp018.austin.ibm.com:8002 - WebSphere Commerce - Administration Console - wcsadmin - Microsoft Internet Explorer
Select ▸
Site Administration Console
Security
Monitoring
Configuration
Store Archives
Help
Logout > Home > Scheduler

Scheduler Status Display

Page Number 1 Go

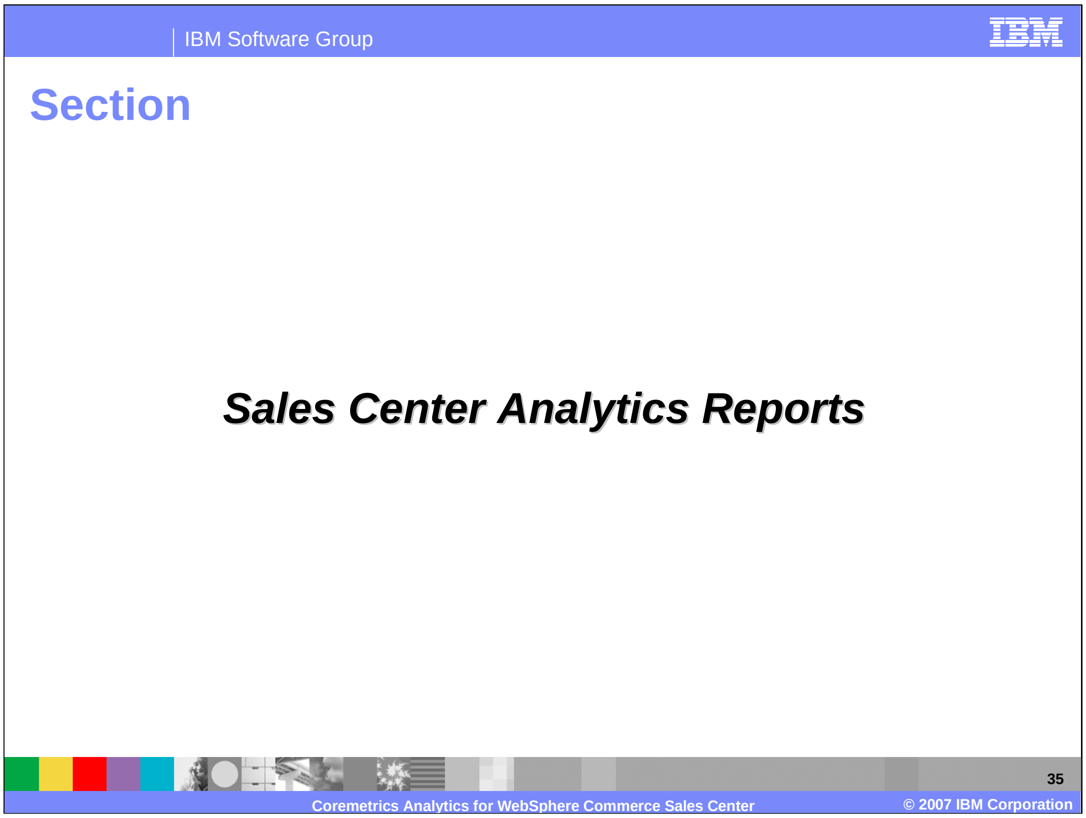
65 items

« First | 1 of 5 | Next ▸ | Last »

Date range: 4/3/07 12:00 AM

□	Command	State	Status	Start date	Finished	Application Type
□	SendEmailActivity	Inactive		4/2/07 5:19 PM		
□	SendEmailActivity	Inactive		4/2/07 7:25 PM		
□	RaiseECEvent	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	CleanECEvent	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	ExperimentCheckExpiry	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	ScheduledContentManagedFileEARUpdate	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	DynaCacheInvalidation	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	SendTransactedMsg	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	PaySynchronizePM	Complete	Successful	4/3/07 12:00 AM	4/3/07 12:00 AM	
□	RaiseECEvent	Complete	Successful	4/3/07 12:05 AM	4/3/07 12:05 AM	
□	SendTransactedMsg	Complete	Successful	4/3/07 12:05 AM	4/3/07 12:05 AM	
□	PaySynchronizePM	Complete	Successful	4/3/07 12:05 AM	4/3/07 12:05 AM	
□	RaiseECEvent	Complete	Successful	4/3/07 12:10 AM	4/3/07 12:10 AM	
□	DynaCacheInvalidation	Complete	Successful	4/3/07 12:10 AM	4/3/07 12:10 AM	
□	SendTransactedMsg	Complete	Successful	4/3/07 12:10 AM	4/3/07 12:10 AM	

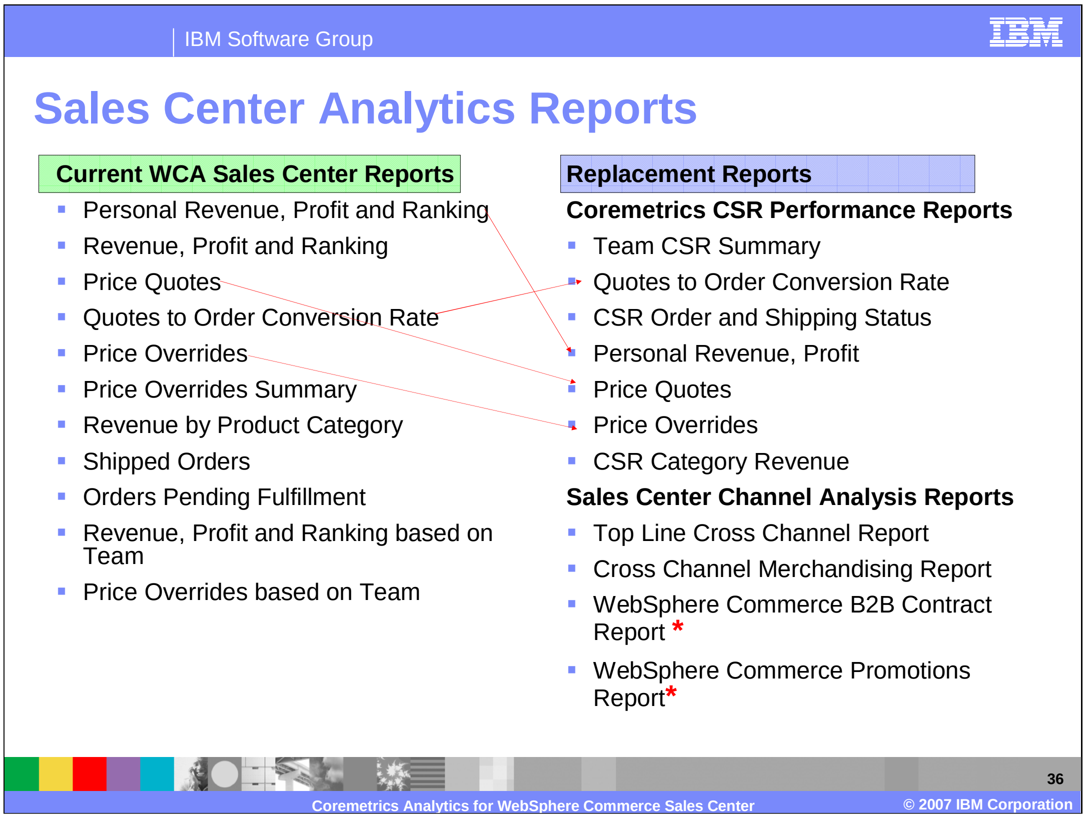
The status of the scheduled job run, indicated by the **RaiseECEvent** command in the screen capture, will be available in the administrative console so that administrators can see if the job has run successfully and transmitted data to Coremetrics.



Section

Sales Center Analytics Reports

This section discusses the two new Sales Center Analytic Reports.



IBM Software Group

Sales Center Analytics Reports

Current WCA Sales Center Reports

- Personal Revenue, Profit and Ranking
- Revenue, Profit and Ranking
- Price Quotes
- Quotes to Order Conversion Rate
- Price Overrides
- Price Overrides Summary
- Revenue by Product Category
- Shipped Orders
- Orders Pending Fulfillment
- Revenue, Profit and Ranking based on Team
- Price Overrides based on Team

Replacement Reports

Coremetrics CSR Performance Reports

- Team CSR Summary
- Quotes to Order Conversion Rate
- CSR Order and Shipping Status
- Personal Revenue, Profit
- Price Quotes
- Price Overrides
- CSR Category Revenue

Sales Center Channel Analysis Reports

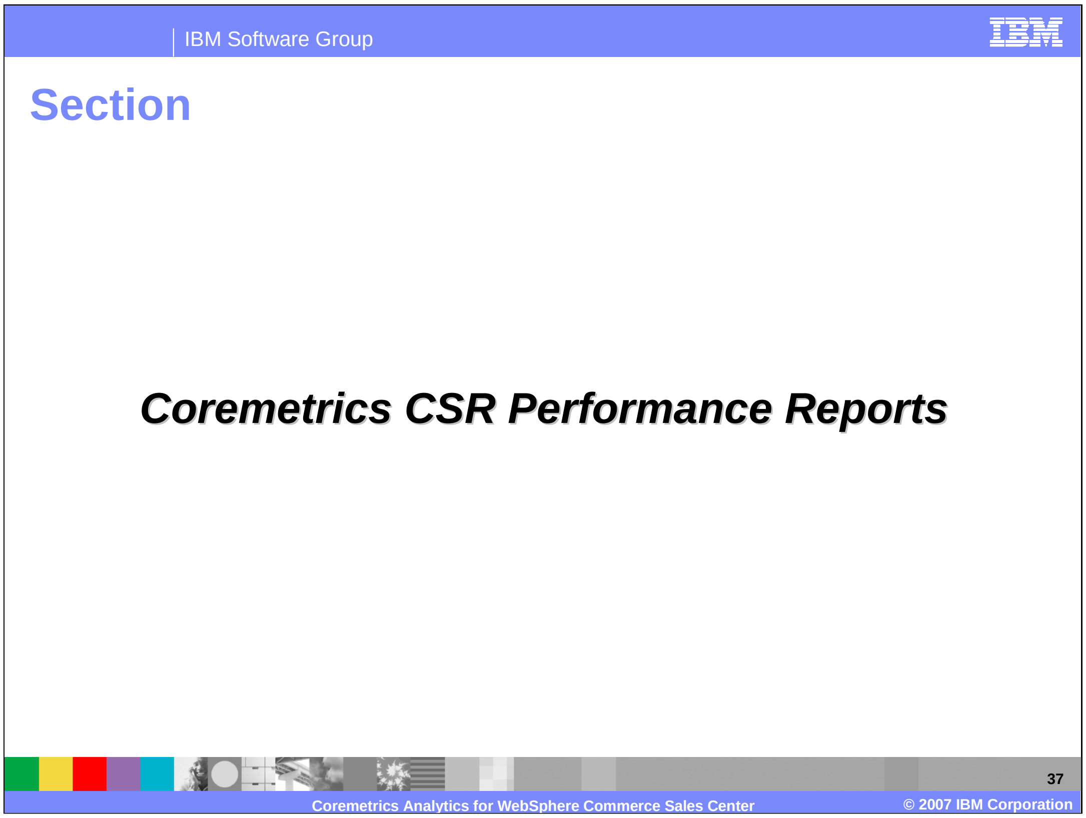
- Top Line Cross Channel Report
- Cross Channel Merchandising Report
- WebSphere Commerce B2B Contract Report *
- WebSphere Commerce Promotions Report*

36

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

Currently, the reports on the left are supported by WebSphere Commerce Analyzer. With the enablement of Coremetrics for Sales Center analytics, the following reports on the right will act as replacements for those reports in the WebSphere Commerce Analyzer. Note that the replacement reports for CSR performance are Coremetrics reports, not WCA reports. There will be no migration path for WCA customers to move their data mart to the Coremetrics data mart. If historical data is required, it is left to customers to negotiate a one-time import process with Coremetrics. The last two reports for the Sales Center Channel analysis are existing reports and with the transfer of Sales Center data these reports will include that information also.



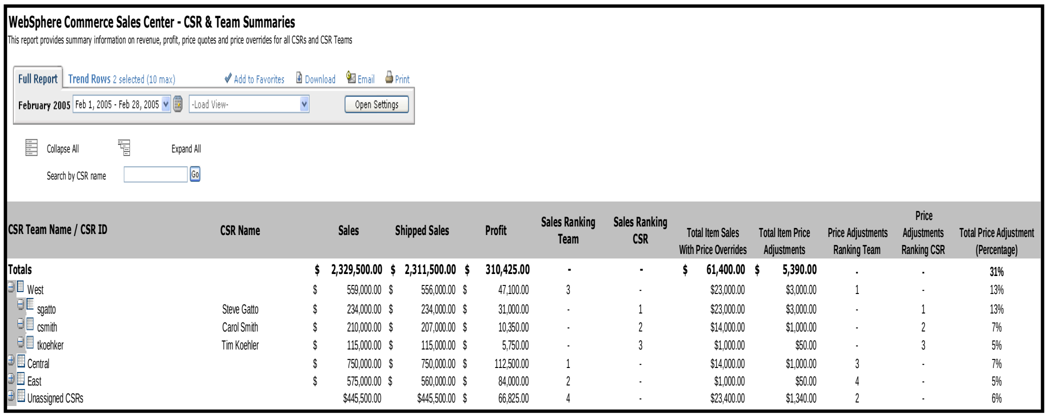
Section

Coremetrics CSR Performance Reports

This section of the presentation looks at the Coremetrics CSR Performance Reports.

Team CSR Summary Report

- Provides revenue, profit and price overrides for all CSR’s
- Replaces current Price Overrides and Revenue, profit and ranking reports



WebSphere Commerce Sales Center - CSR & Team Summaries

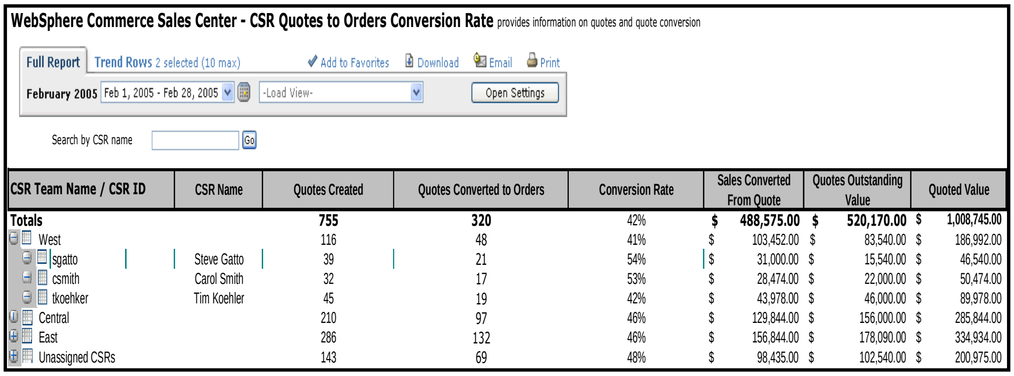
This report provides summary information on revenue, profit, price quotes and price overrides for all CSRs and CSR Teams

CSR Team Name / CSR ID	CSR Name	Sales	Shipped Sales	Profit	Sales Ranking Team	Sales Ranking CSR	Total Item Sales With Price Overrides	Total Item Price Adjustments	Price Adjustments Ranking Team	Price Adjustments Ranking CSR	Total Price Adjustment (Percentage)
Totals		**$ 2,329,500.00**	**$ 2,311,500.00**	**$ 310,425.00**	**-**	**-**	**$ 61,400.00**	**$ 5,390.00**	**-**	**-**	**31%**
West		$ 559,000.00	$ 556,000.00	$ 47,100.00	3	-	$23,000.00	$3,000.00	1	-	13%
sgatto	Steve Gatto	$ 234,000.00	$ 234,000.00	$ 31,000.00	-	1	$23,000.00	$3,000.00	-	1	13%
csmith	Carol Smith	$ 210,000.00	$ 207,000.00	$ 10,350.00	-	2	$14,000.00	$1,000.00	-	2	7%
tkoehker	Tim Koehler	$ 115,000.00	$ 115,000.00	$ 5,750.00	-	3	$1,000.00	$50.00	-	3	5%
Central		$ 750,000.00	$ 750,000.00	$ 112,500.00	1	-	$14,000.00	$1,000.00	3	-	7%
East		$ 575,000.00	$ 560,000.00	$ 84,000.00	2	-	$1,000.00	$50.00	4	-	5%
Unassigned CSRs		$445,500.00	$445,500.00	$ 66,825.00	4	-	$23,400.00	$1,340.00	2	-	6%

The Team CSR Summary report provides summary information on revenue, profit and price overrides for all CSR’s and CSR Teams. When the CSR manager selects this report it will load with data from all CSR teams with the hierarchy collapsed. The CSR Manager can expand the hierarchy to see the names of individuals on each team. The CSR manager can search on an individual CSR’s name.

Quotes to Order Conversion Rate report

- Provides information on quotes and quote conversions
- Replaces current Quotes to Order CSR report



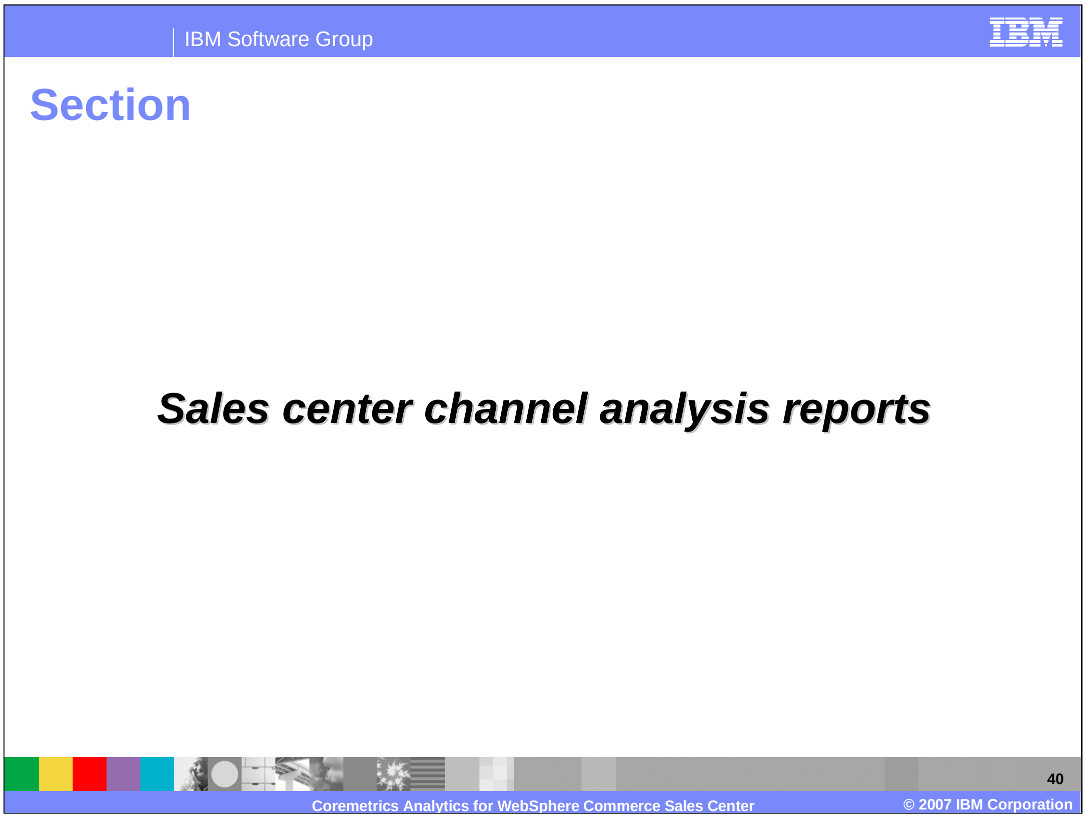
WebSphere Commerce Sales Center - CSR Quotes to Orders Conversion Rate provides information on quotes and quote conversion

February 2005 Feb 1, 2005 - Feb 28, 2005

Search by CSR name

CSR Team Name / CSR ID	CSR Name	Quotes Created	Quotes Converted to Orders	Conversion Rate	Sales Converted From Quote	Quotes Outstanding Value	Quoted Value
Totals		**755**	**320**	42%	**$ 488,575.00**	**$ 520,170.00**	$ 1,008,745.00
West		116	48	41%	$ 103,452.00	$ 83,540.00	$ 186,992.00
sgatto	Steve Gatto	39	21	54%	$ 31,000.00	$ 15,540.00	$ 46,540.00
csmith	Carol Smith	32	17	53%	$ 28,474.00	$ 22,000.00	$ 50,474.00
tkoehker	Tim Koehler	45	19	42%	$ 43,978.00	$ 46,000.00	$ 89,978.00
Central		210	97	46%	$ 129,844.00	$ 156,000.00	$ 285,844.00
East		286	132	46%	$ 156,844.00	$ 178,090.00	$ 334,934.00
Unassigned CSRs		143	69	48%	$ 98,435.00	$ 102,540.00	$ 200,975.00

The Quotes to Order Conversion Rate report provides information on quotes and the total number of quotes converted during a time period. When the CSR manager wants to see how many quotes a sales person is creating, they can use this report as an indication of the sales persons ability to move a customer to a product recommendation and price proposal. The report opens with the hierarchy loaded and collapsed at the team level. The CSR manager can expand the hierarchy to find information about a particular CSR. The CSR manager can also search using the search box to find information about a specific CSR.



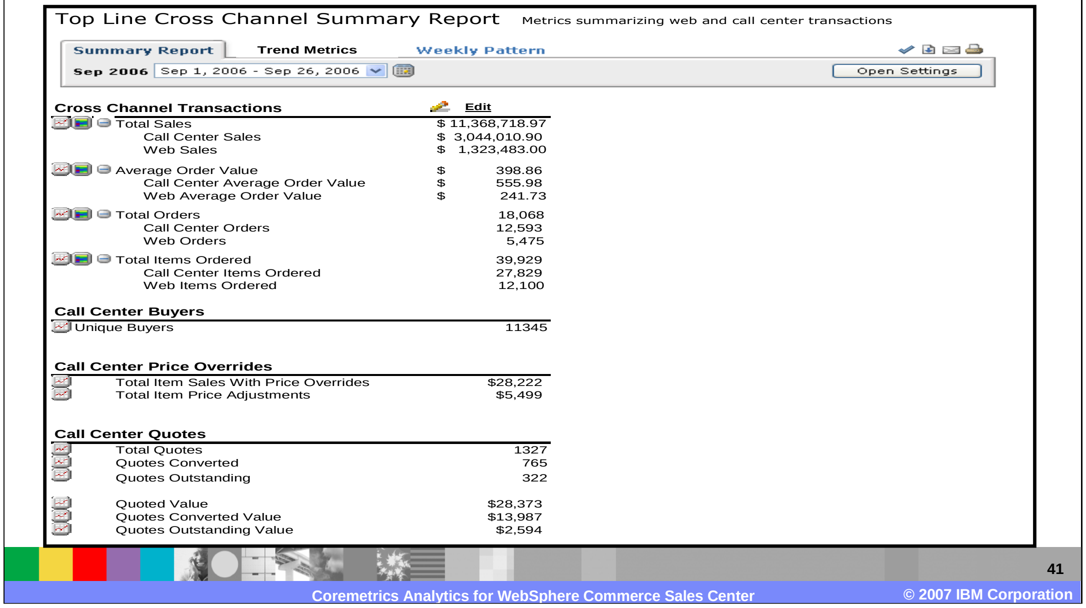
Section

Sales center channel analysis reports

This section of the presentation looks at the Sales Center Channel Analysis Reports.

Top Line Cross Channel Summary Report

- Provides metrics on Web and call center transactions.



Top Line Cross Channel Summary Report Metrics summarizing web and call center transactions

Cross Channel Transactions	
Total Sales	$ 11,368,718.97
Call Center Sales	$ 3,044,010.90
Web Sales	$ 1,323,483.00
Average Order Value	$ 398.86
Call Center Average Order Value	$ 555.98
Web Average Order Value	$ 241.73
Total Orders	18,068
Call Center Orders	12,593
Web Orders	5,475
Total Items Ordered	39,929
Call Center Items Ordered	27,829
Web Items Ordered	12,100

Call Center Buyers	
Unique Buyers	11345

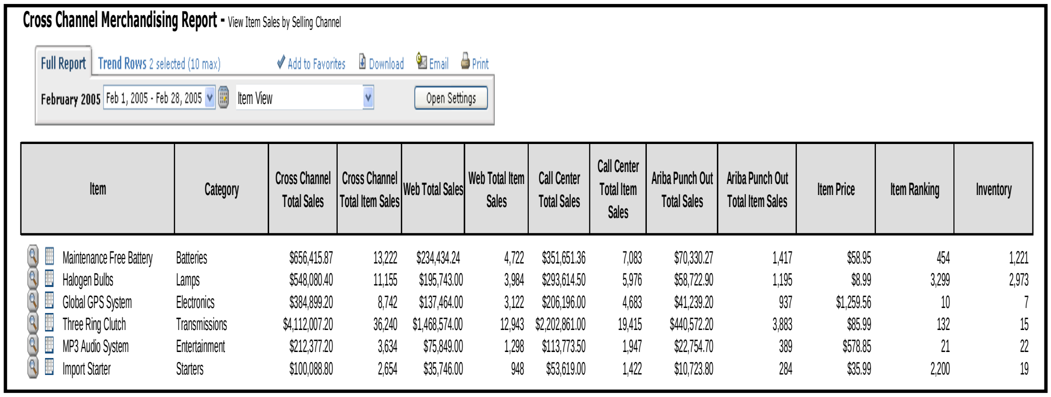
Call Center Price Overrides	
Total Item Sales With Price Overrides	$28,222
Total Item Price Adjustments	$5,499

Call Center Quotes	
Total Quotes	1327
Quotes Converted	765
Quotes Outstanding	322
Quoted Value	$28,373
Quotes Converted Value	$13,987
Quotes Outstanding Value	$2,594

The Top Line Cross Channel Summary Report provides metrics summarizing Web and call center transactions.

Cross Channel Merchandising report

- Provides metrics on item sales by selling channel

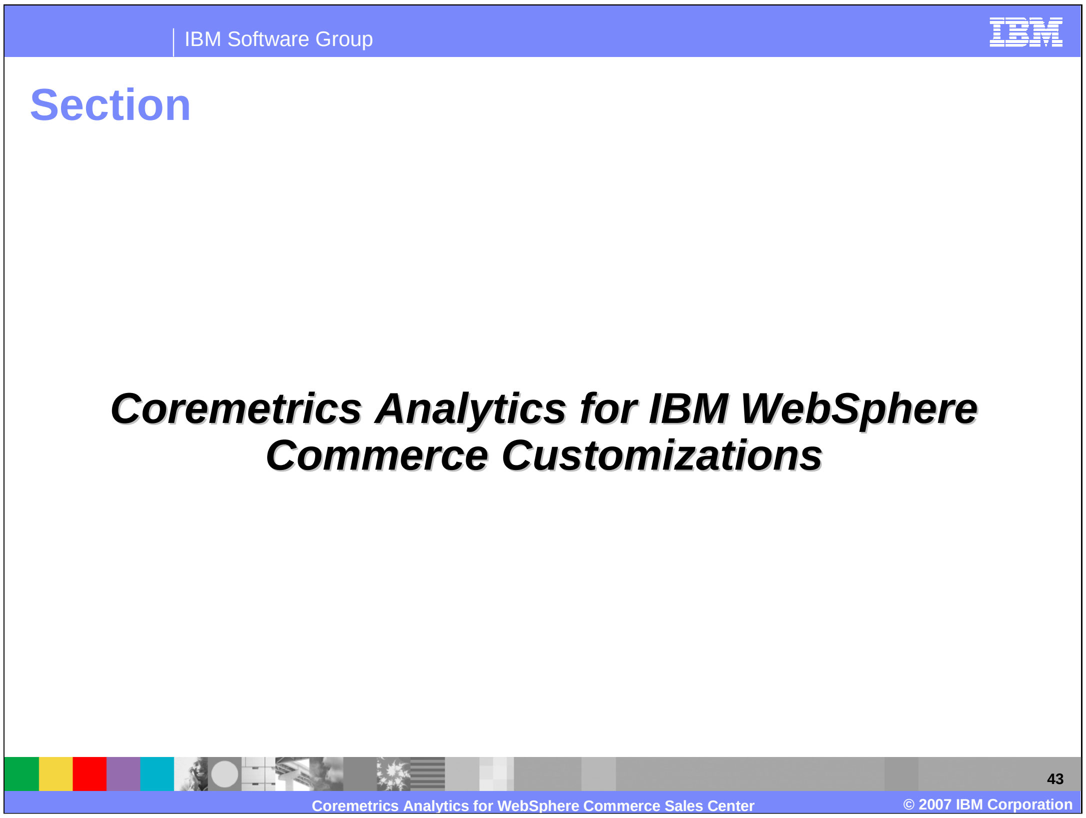


Cross Channel Merchandising Report - View Item Sales by Selling Channel

February 2005 Feb 1, 2005 - Feb 28, 2005 Item View

Item	Category	Cross Channel Total Sales	Cross Channel Total Item Sales	Web Total Sales	Web Total Item Sales	Call Center Total Sales	Call Center Total Item Sales	Ariba Punch Out Total Sales	Ariba Punch Out Total Item Sales	Item Price	Item Ranking	Inventory
Maintenance Free Battery	Batteries	$656,415.87	13,222	$234,434.24	4,722	$351,651.36	7,083	$70,330.27	1,417	$58.95	454	1,221
Halogen Bulbs	Lamps	$548,080.40	11,155	$195,743.00	3,984	$293,614.50	5,976	$58,722.90	1,195	$8.99	3,299	2,973
Global GPS System	Electronics	$384,899.20	8,742	$137,464.00	3,122	$206,196.00	4,683	$41,239.20	937	$1,259.56	10	7
Three Ring Clutch	Transmissions	$4,112,007.20	36,240	$1,468,574.00	12,943	$2,202,861.00	19,415	$440,572.20	3,883	$85.99	132	15
MP3 Audio System	Entertainment	$212,377.20	3,634	$75,849.00	1,298	$113,773.50	1,947	$22,754.70	389	$578.85	21	22
Import Starter	Starters	$100,088.80	2,654	$35,746.00	948	$53,619.00	1,422	$10,723.80	284	$35.99	2,200	19

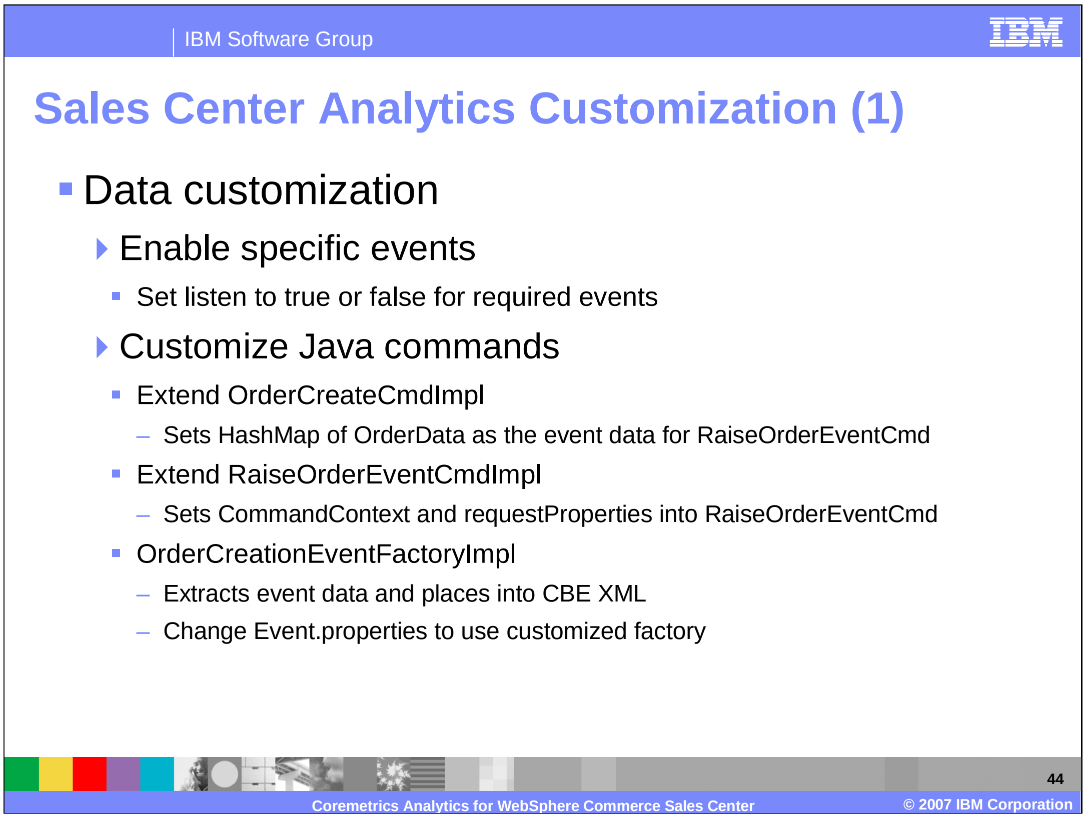
The Cross Channel Merchandising Report provides metrics on items sales by selling channel.



Section

Coremetrics Analytics for IBM WebSphere Commerce Customizations

This section discusses what it takes to customize the Coremetrics Analytics for WebSphere Commerce Sales Center feature.

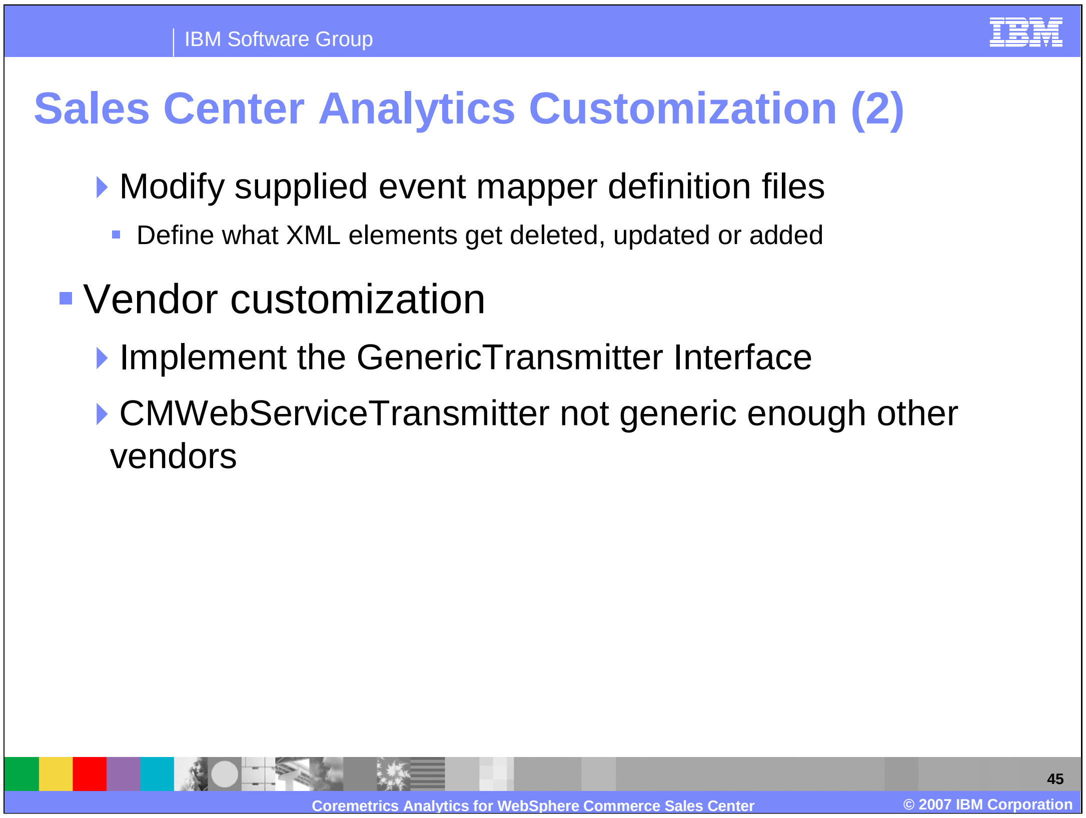


Sales Center Analytics Customization (1)

- Data customization
  - Enable specific events
    - Set listen to true or false for required events
  - Customize Java commands
    - Extend OrderCreateCmdImpl
      - Sets HashMap of OrderData as the event data for RaiseOrderEventCmd
    - Extend RaiseOrderEventCmdImpl
      - Sets CommandContext and requestProperties into RaiseOrderEventCmd
    - OrderCreationEventFactoryImpl
      - Extracts event data and places into CBE XML
      - Change Event.properties to use customized factory

There are two customization points within the Coremetrics Analytics for WebSphere Commerce Sales Center feature and they are data customization and vendor customization.

Customization of the data sent to Coremetrics (or some other provider) can be achieved in three ways. The first is to edit wc-server.xml and specify exactly which supported events should be enabled. This can be done be setting the “event listen” tag to true or false for each event. The second is to customize the Java commands that generate the original Commerce events. For example, to add more data into the OrderCreate event, a customer can extend: OrderCreateCmdImpl, RaiseOrderEventCmdImpl or OrderCreationEventFactoryImpl.

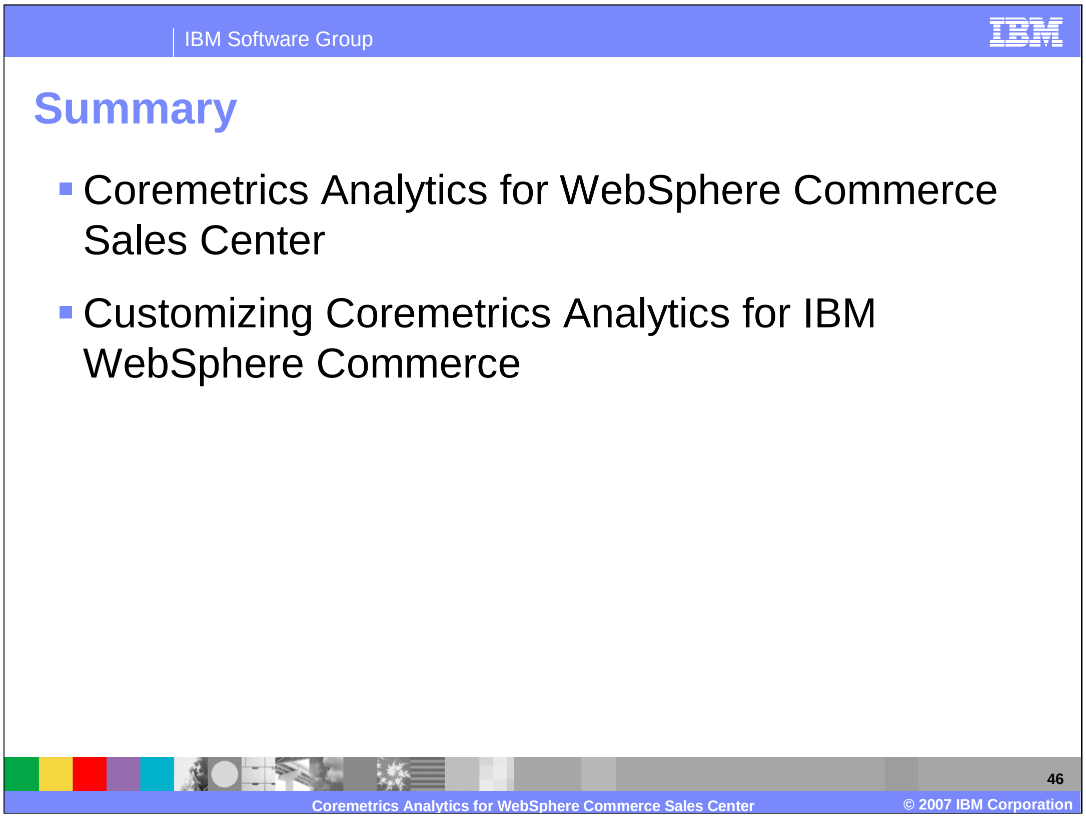


Sales Center Analytics Customization (2)

- Modify supplied event mapper definition files
  - Define what XML elements get deleted, updated or added
- Vendor customization
  - Implement the GenericTransmitter Interface
  - CMWebServiceTransmitter not generic enough other vendors

The third method of customization is to modify the supplied event mapper definition files. Since these files define what transformations occur on the XML files, what XML elements get deleted, updated or added, the end resultant XML can be modified to a customer’s needs.

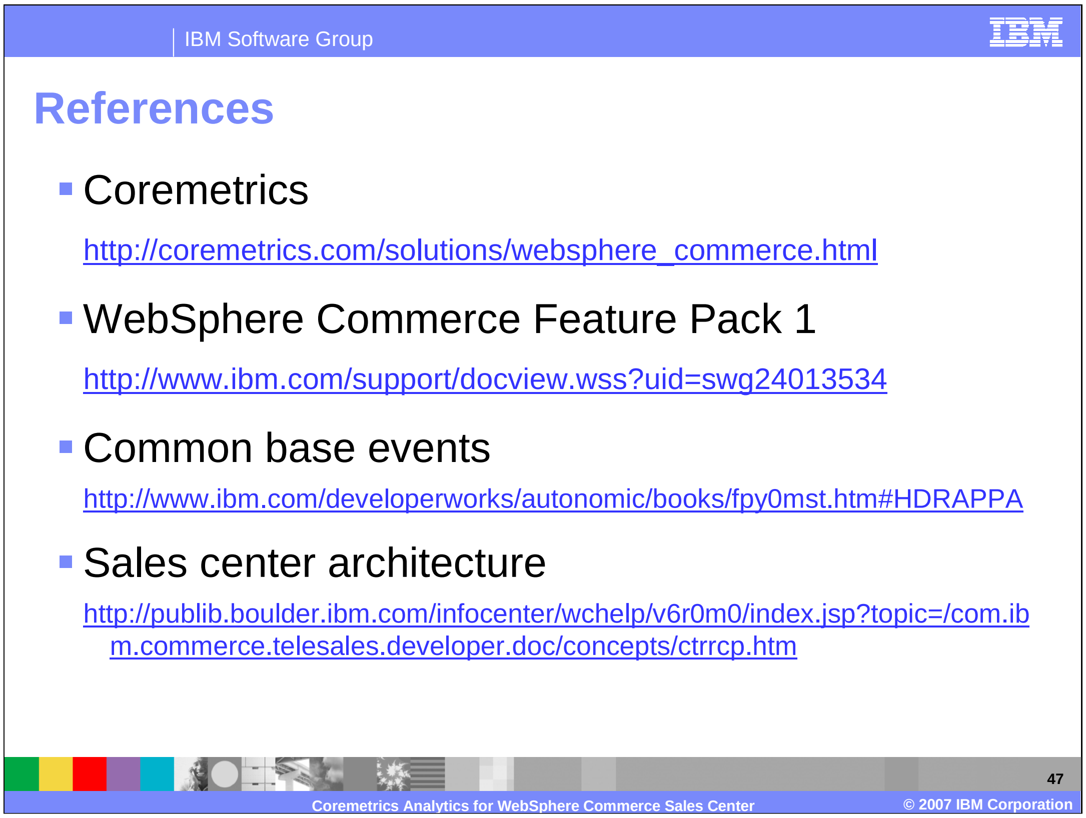
For vendor customization, the event/data transmission to another vendor is achieved by implementing the GenericTransmitter interface. The implementation will send the generated events to any host by any method - like Web service, FTP, e-mail, printer and so on - as required. The CMWebServiceTransmitter is not generic enough to be reused for another vendor. For example, inputs like the Coremetrics client ID are probably not applicable for other vendors - they may have some other identifier key or some other user authentication mechanism.



Summary

- Coremetrics Analytics for WebSphere Commerce Sales Center
- Customizing Coremetrics Analytics for IBM WebSphere Commerce

In this presentation, you learned about the details of the Coremetrics Analytics for WebSphere Commerce Sales Center feature and what it takes to customize this feature.



IBM Software Group

References

- Coremetrics

  http://coremetrics.com/solutions/websphere_commerce.html

- WebSphere Commerce Feature Pack 1

  http://www.ibm.com/support/docview.wss?uid=swg24013534

- Common base events

  http://www.ibm.com/developerworks/autonomic/books/fpy0mst.htm#HDRAPPA

- Sales center architecture

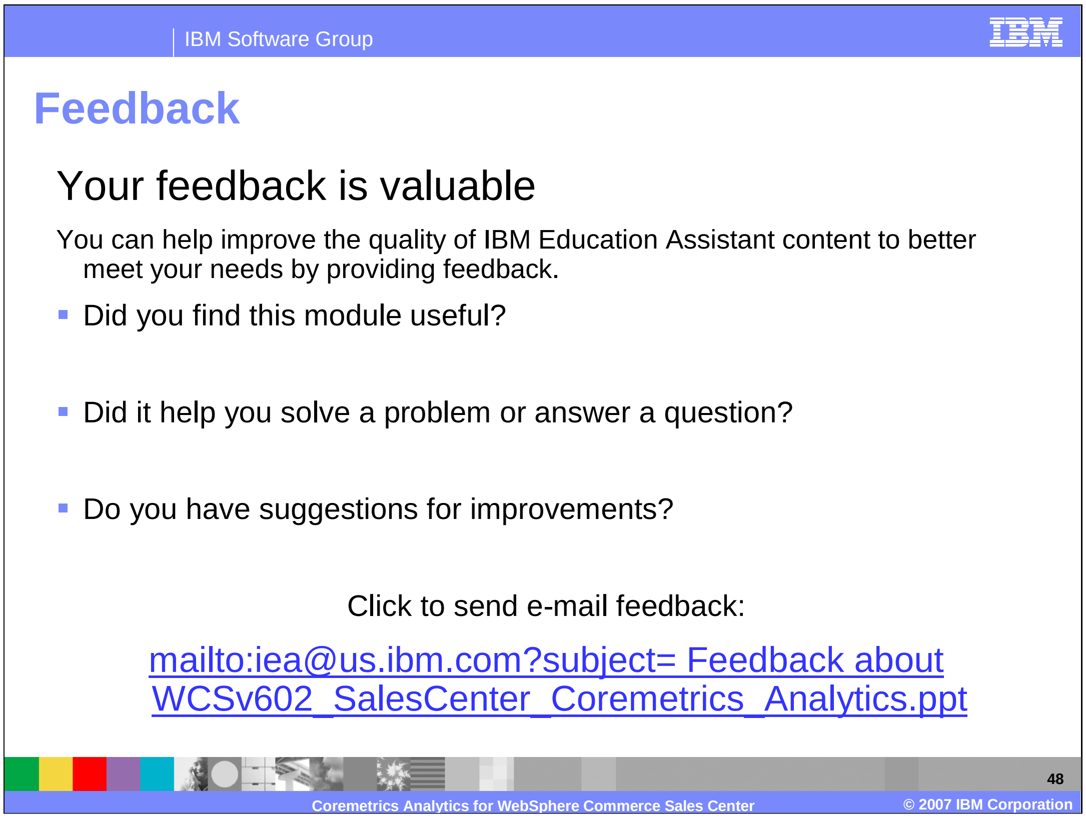
  http://publib.boulder.ibm.com/infocenter/wchelp/v6r0m0/index.jsp?topic=/com.ibm.commerce.telesales.developer.doc/concepts/ctrrcp.htm

47

Coremetrics Analytics for WebSphere Commerce Sales Center

© 2007 IBM Corporation

For more information regarding Coremetrics, WebSphere Commerce Feature Pack 1, Common Base Events or the Sales center architecture, visit the sites indicated in the presentation.



Feedback

Your feedback is valuable

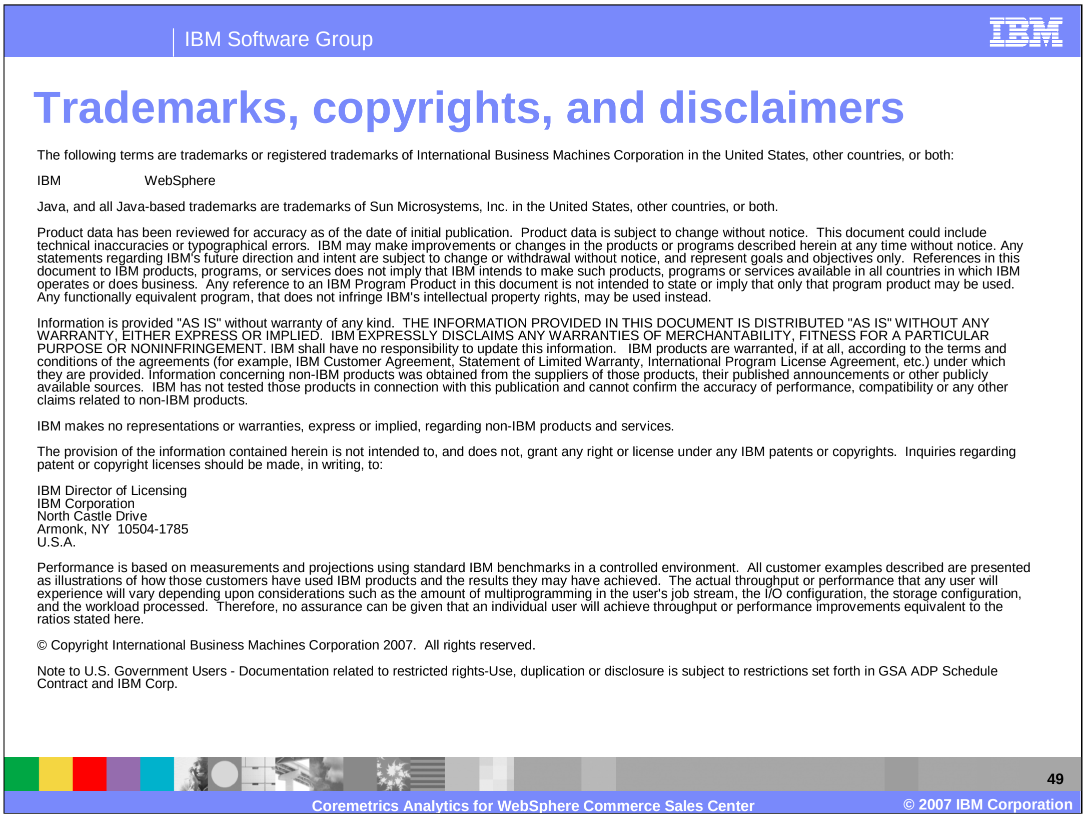
You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

- 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 WCSv602_SalesCenter_Coremetrics_Analytics.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:

IBM	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.

Note to U.S. Government Users - Documentation related to restricted rights-Use, duplication or disclosure is subject to restrictions set forth in GSA ADP Schedule Contract and IBM Corp.