



IBM Software Group

WebSphere® Business Monitor V6.1

XSD events



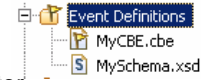
@business on demand.

© 2008 IBM Corporation
Updated February 22, 2008

This presentation introduces using the new XSD style events in WebSphere Business Monitor Version 6.1.

XSD event definitions

- Uses XSDs to describe event business data
- Retain compatibility with 6.0.2-style common base event definitions
- Continue to use common base event for event envelope format at runtime.
- XSD event definitions are shown alongside common base event definitions in the 'Event Definitions' group on the Project Explorer
 - ▶ The default editor in WebSphere Integration Developer is the BO editor
 - ▶ The default editor in a Rational® Application Developer environment is the XSD editor



▼ Event Type Details (Read only, managed by application)

Specify the event type or the XML schemas that together describe the structure of this inbound event. You can specify an extension name, event parts, or both.

Extension name:

Event parts:

ID	Name	Type	Path
BaseData	BaseData	wbi:Event	cbe:CommonBaseEv
BPELData	BPELData	bpc:BPC.BFM.ACTIVITY.M...	cbe:CommonBaseEv
Input	Input	bo:Order	cbe:CommonBaseEv



A new event definition is used to define the business payload in events destined for Monitor. The new events use schema definitions to describe the layout of the payload; however the event itself is still using the common base event envelope as a wrapper for the event. At runtime, XML is used to represent the business object based on the XSD for the business object.

Both the older style events and the new style events are still supported, so in the monitor model editor you will see both types of definitions listed. The default editor in WebSphere Integration Developer is the business object editor, but in Rational Application Developer the default editor is the XSD editor.

In the screen print you can see the event type details for an inbound event which is based on an XSD style event definition. There are three event parts, including base event data, BPEL data, and the payload. Each part has its own schema definition.

Event payload

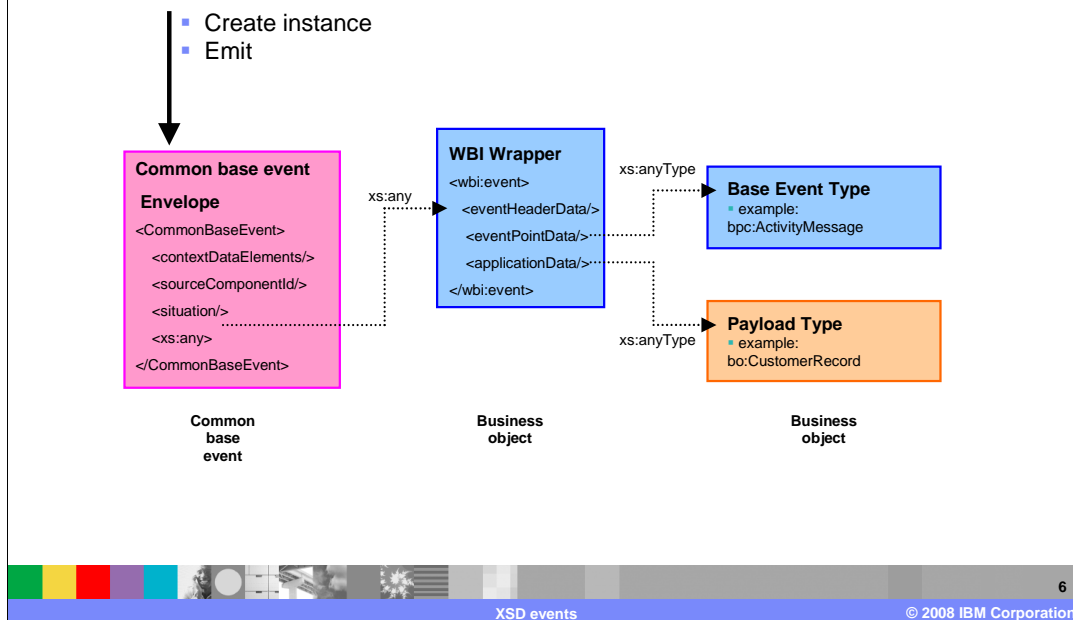
- Treat common base event strictly as a top level “transport” envelope
 - ▶ Only support common event fields required for coexistence or correct functioning
- Move component event payload from extended data section to xs:any field
 - ▶ No longer shred the event payload – extendedDataElements not used
- Construct event data as service data object in xs:any slot

```
<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" xmlns:cbe="http://www.ibm.com/AC/commonbaseevent1_0_1"
  targetNamespace="http://www.ibm.com/AC/commonbaseevent1_0_1" version="1.0.1" elementFormDefault="qualified">
  <xsd:complexType name="CommonBaseEventType">
    <xsd:sequence>
      <xsd:element name="contextDataElements" type="cbe:ContextDataElementType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="extendedDataElements" type="cbe:ExtendedDataElementType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="associatedEvents" type="cbe:AssociatedEventType" minOccurs="0" maxOccurs="unbounded" />
      <xsd:element name="reporterComponentId" type="cbe:ComponentIdentificationType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="sourceComponentId" type="cbe:ComponentIdentificationType" minOccurs="1" maxOccurs="1" />
      <xsd:element name="msgDataElement" type="cbe:MsgDataElementType" minOccurs="0" maxOccurs="1" />
      <xsd:element name="situation" type="cbe:Situation" minOccurs="1" maxOccurs="1" />
      <xsd:any namespace="##other" minOccurs="0" maxOccurs="unbounded" processContents="skip" />
    </xsd:sequence>
  </xsd:complexType>
</xsd:schema>
```



In the new event format, the common base event format is still used as the wrapper for the event data, however the event payload is stored in the xsd:any slot instead of the extendedDataElements section. Also the data is stored in XML notation rather than shredded event notation. Here you see the schema for the CommonBaseEventType, and in it you see the extendedDataElements section which is not used, and highlighted in red is the xsd:any slot where you will now find the payload.

XML event format



When an event is emitted in the new format, the common base event envelope is used as a wrapper for the event. The `xs:any` slot is used to store the business object data. The actual XML name will be `wbi:event` for the WBI payload wrapper. Within the `wbi:event` data, you will find `eventPointData` which is used to define the type of event, and the `applicationData` which is used to store the payload.

Common base event browser – Event part 1

globalInstanceid	CE1F2D1704DD978F0A1DC66D46F2F6830
extensionName	BPC.BFM.ACTIVITY.MESSAGING
localInstanceid	
creationTime	2007-09-19T17:19:09.875Z
severity	10
msg	
priority	
sequenceNumber	11
repeatCount	
elapsedTime	
contextDataElement / WBIEventVersion / contextValue	6.1
contextDataElement / WBISESSION_ID / contextValue	192.168.253.128:testModule::operation1.1190222349781.2136637274
contextDataElement / ECSCurrentID / contextValue	_AI-10040115.1e7c5d5.762573f.db0026
contextDataElement / ECSParentID / contextValue	_PI-10030115.1e7c5d5.762573f.db0026
reporterComponentId	
sourceComponentId / component	WPSWPlatform 6.1 [ND 6.1.0.9 c#90722.41] [WBI 6.1.0.0 e0736.02] [WBM 6.1.0.0 e0736.23]
sourceComponentId / subComponent	BFM
sourceComponentId / componentIdType	ProductName
sourceComponentId / instanceid	WBMonSrv_wps_CellWBMonSrv_wps_Node/server1
sourceComponentId / application	
sourceComponentId / executionEnvironment	Windows XP[86]#5.1 build 2600 Service Pack 2
sourceComponentId / location	192.168.253.128
sourceComponentId / locationType	Hostname
sourceComponentId / processId	1040
sourceComponentId / threadId	WebContainer : 1
sourceComponentId / componentType	http://www.ibm.com/namespaces/autonomic/Workflow_Engine
msgDataElement	
situation / categoryName	StopSituation
situation / situationType / reasoningScope	EXTERNAL
situation / StopSituation / successDisposition	SUCCESSFUL
situation / StopSituation / situationQualifier	STOP_COMPLETED
wbi:event	<wbi:event xmlns:wbi="http://www.ibm.com/xmlns/prod/websphere/monitoring8.1" xmlns:psi="http://www...



This is a screen print of the first part of an XSD style event as seen in the common base event browser. Note that extensionName identifies the event type just like in 6.0.2. Note the absence of extendedDataElements. The xsd:any slot is always populated with the wbi:event element, which contains the event payload. In the common base event browser, you can click on the link in the second column on the row for wbi:event, then it will display the XML for the payload. This is shown in the next slide.

IBM Software Group IBM

Common base event browser – Event part 2

situation / StopSituation / situationQualifier	STOP_COMPLETED
wbi:event	<wbi:event xmlns:wbi="http://www.ibm.com/xmlns/prod/websphere/monitoring/8.1" xmlns:xsi="http://www...

wbi:event
Show the details for the Common Base Event path wbi:event

↓

expand collapse

```

<wbi:event>
  <wbi:eventHeaderData>
    <wbi:WBISESSION_ID> 192.168.253.128;testModule;operation 1;1190222349781;2136637274</wbi:WBISESSION_ID>
    <wbi:ECSCurrentID> _AI:10040115.1ec7c5d5.7f02573f.db0026</wbi:ECSCurrentID>
    <wbi:ECSParentID> _PI:10030115.1ec7c5d5.7f02573f.db0020</wbi:ECSParentID>
    <wbi:WBIEventVersion> 6.1</wbi:WBIEventVersion>
  </wbi:eventHeaderData>
  <wbi:eventPointData xsi:type="bpc:BPC.BFM.ACTIVITY.MESSAGE" >
    <wbi:eventNature> EXIT</wbi:eventNature>
    <wbi:payloadType> full</wbi:payloadType>
    <bpc:BPCEventCode> 21011</bpc:BPCEventCode>
    <bpc:processTemplateName> testProcess</bpc:processTemplateName>
    <bpc:processTemplateValidFrom> Wed 2007-08-01 19:11:01.000</bpc:processTemplateValidFrom>
    <bpc:activityKind> 23 - KIND_RECEIVE</bpc:activityKind>
    <bpc:state> 5 - STATE_FINISHED</bpc:state>
    <bpc:bpelId> 4</bpc:bpelId>
    <bpc:activityTemplateName> Receive</bpc:activityTemplateName>
    <bpc:activityTemplateId> _AT:90020115.1e774792.7f02573f.db000a</bpc:activityTemplateId>
    <bpc:principal> UNAUTHENTICATED</bpc:principal>
  </wbi:eventPointData>
  <wbi:applicationData>
    <wbi:content wbi:name="input1" >
      <wbi:value xsi:type="xsd:string" > aabcc</wbi:value>
    </wbi:content>
  </wbi:applicationData>
</wbi:event>

```

Close

8

XSD events © 2008 IBM Corporation

In the common base event browser, when you click on the link in the second column on the row for wbi:event, it displays the XML for the payload.

The eventHeaderData is the session correlation information which is used for process monitoring only.

The eventPointData contains the base event information emitted by the runtime event point. So you see the message name, event nature, state, BPEL id, template name and other information. In this example it is a type corresponding to type BPC.BFM.ACTIVITY.MESSAGE.

The optional applicationData section contains any payload data for the event. In this example you see one string which is the payload corresponding to the argument for a BPEL receive activity.

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_WBMonitorV61_XSD_Events.ppt

This module is also available in PDF format at: ..\\WBMonitorV61_XSD_Events.pdf



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 Rational WebSphere

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.

© Copyright International Business Machines Corporation 2008. All rights reserved.

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.

