



# BEA WebLogic Workshop™ Help

Version 8.1 SP2  
November 2003

# Copyright

Copyright © 2003 BEA Systems, Inc. All Rights Reserved.

## Restricted Rights Legend

This software and documentation is subject to and made available only pursuant to the terms of the BEA Systems License Agreement and may be used or copied only in accordance with the terms of that agreement. It is against the law to copy the software except as specifically allowed in the agreement. This document may not, in whole or in part, be copied, photocopied, reproduced, translated, or reduced to any electronic medium or machine readable form without prior consent, in writing, from BEA Systems, Inc.

Use, duplication or disclosure by the U.S. Government is subject to restrictions set forth in the BEA Systems License Agreement and in subparagraph (c)(1) of the Commercial Computer Software–Restricted Rights Clause at FAR 52.227–19; subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.227–7013, subparagraph (d) of the Commercial Computer Software—Licensing clause at NASA FAR supplement 16–52.227–86; or their equivalent.

Information in this document is subject to change without notice and does not represent a commitment on the part of BEA Systems. THE SOFTWARE AND DOCUMENTATION ARE PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND INCLUDING WITHOUT LIMITATION, ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. FURTHER, BEA Systems DOES NOT WARRANT, GUARANTEE, OR MAKE ANY REPRESENTATIONS REGARDING THE USE, OR THE RESULTS OF THE USE, OF THE SOFTWARE OR WRITTEN MATERIAL IN TERMS OF CORRECTNESS, ACCURACY, RELIABILITY, OR OTHERWISE.

## Trademarks or Service Marks

BEA, Jolt, Tuxedo, and WebLogic are registered trademarks of BEA Systems, Inc. BEA Builder, BEA Campaign Manager for WebLogic, BEA eLink, BEA Liquid Data for WebLogic, BEA Manager, BEA WebLogic Commerce Server, BEA WebLogic Enterprise, BEA WebLogic Enterprise Platform, BEA WebLogic Enterprise Security, BEA WebLogic Express, BEA WebLogic Integration, BEA WebLogic Personalization Server, BEA WebLogic Platform, BEA WebLogic Portal, BEA WebLogic Server, BEA WebLogic Workshop and How Business Becomes E–Business are trademarks of BEA Systems, Inc.

All other trademarks are the property of their respective companies.

# Table of Contents

<b>Java Class Reference.....</b>	<b>1</b>
<b>com.bea.control Package.....</b>	<b>2</b>
<b>com.bea.control B2BCallback Interface.....</b>	<b>3</b>
<b>com.bea.control B2BCallbackControl Interface.....</b>	<b>4</b>
<b>com.bea.control EBXMLControl Interface.....</b>	<b>5</b>
<b>com.bea.control EBXMLControl.Callback Interface.....</b>	<b>7</b>
<b>com.bea.control EmailControl Interface.....</b>	<b>8</b>
<b>com.bea.control FileControl Interface.....</b>	<b>13</b>
<b>com.bea.control HttpControl Interface.....</b>	<b>22</b>
<b>com.bea.control MQControl Interface.....</b>	<b>27</b>
<b>com.bea.control ProcessControl Interface.....</b>	<b>33</b>
<b>com.bea.control ProcessControlException Class.....</b>	<b>37</b>
<b>com.bea.control PublishControl Interface.....</b>	<b>39</b>
<b>com.bea.control RosettaNetControl Interface.....</b>	<b>41</b>
<b>com.bea.control RosettaNetControl.Callback Interface.....</b>	<b>44</b>
<b>com.bea.control ServiceBrokerControl Interface.....</b>	<b>45</b>
<b>com.bea.control SubscriptionControl Interface.....</b>	<b>47</b>
<b>com.bea.control SubscriptionControl.Callback Interface.....</b>	<b>49</b>
<b>com.bea.control TaskControl Interface.....</b>	<b>50</b>
<b>com.bea.control TaskControl.Callback Interface.....</b>	<b>63</b>
<b>com.bea.control TaskWorkerControl Interface.....</b>	<b>65</b>
<b>com.bea.control TPMControl Interface.....</b>	<b>81</b>
<b>com.bea.control TPMControlException Class.....</b>	<b>89</b>

# Table of Contents

<b>com.bea.control WliJMSControl Interface.....</b>	<b>91</b>
<b>com.bea.control WliJMSControl.Callback Interface.....</b>	<b>93</b>
<b>com.bea.data Package.....</b>	<b>94</b>
<b>com.bea.data MessageAttachment Interface.....</b>	<b>95</b>
<b>com.bea.data MessageAttachment.Factory Class.....</b>	<b>98</b>
<b>com.bea.data RawData Interface.....</b>	<b>100</b>
<b>com.bea.data RawData.Factory Class.....</b>	<b>103</b>
<b>com.bea.data RawDataRuntimeException Class.....</b>	<b>106</b>
<b>com.bea.jpd Package.....</b>	<b>108</b>
<b>com.bea.jpd JpdContext Interface.....</b>	<b>109</b>
<b>com.bea.jpd JpdContext.Callback Interface.....</b>	<b>112</b>
<b>com.bea.jpd JpdContext.ExceptionInfo Interface.....</b>	<b>113</b>
<b>com.bea.jpd ProcessDefinition Interface.....</b>	<b>116</b>
<b>com.bea.transform Package.....</b>	<b>117</b>
<b>com.bea.transform TransformException Class.....</b>	<b>118</b>
<b>com.bea.transform TransformFaultCodes Interface.....</b>	<b>124</b>
<b>com.bea.transform TransformSource Interface.....</b>	<b>143</b>
<b>com.bea.wli.bpm.proxy Package.....</b>	<b>144</b>
<b>com.bea.wli.bpm.proxy JpdProxy Class.....</b>	<b>145</b>
<b>com.bea.wli.bpm.proxy JpdProxy.ContextHandler Interface.....</b>	<b>148</b>
<b>com.bea.wli.bpm.proxy JpdProxyException Class.....</b>	<b>150</b>
<b>com.bea.wli.bpm.proxy JpdProxySession Interface.....</b>	<b>152</b>
<b>com.bea.wli.calendar.api Package.....</b>	<b>154</b>

# Table of Contents

<b>com.bea.wli.calendar.api BusinessCalendar Interface.....</b>	<b>155</b>
<b>com.bea.wli.calendar.api BusinessCalendarException Class.....</b>	<b>169</b>
<b>com.bea.wli.calendar.api BusinessCalendarFactory Class.....</b>	<b>171</b>
<b>com.bea.wli.calendar.api BusinessCalendarHome Interface.....</b>	<b>173</b>
<b>com.bea.wli.management.archiving Package.....</b>	<b>174</b>
<b>com.bea.wli.management.archiving ArchivingPolicy Class.....</b>	<b>175</b>
<b>com.bea.wli.management.archiving DocumentDataType Interface.....</b>	<b>178</b>
<b>com.bea.wli.management.archiving TrackingEventType Interface.....</b>	<b>180</b>
<b>com.bea.wli.management.archiving WLIArchiverException Class.....</b>	<b>184</b>
<b>com.bea.wli.management.configuration Package.....</b>	<b>186</b>
<b>com.bea.wli.management.configuration B2BBaseMBean Interface.....</b>	<b>188</b>
<b>com.bea.wli.management.configuration B2BConfigMBean Interface.....</b>	<b>192</b>
<b>com.bea.wli.management.configuration B2BConfigurationException Class.....</b>	<b>198</b>
<b>com.bea.wli.management.configuration BPMConfigurationMBean Interface.....</b>	<b>200</b>
<b>com.bea.wli.management.configuration BusinessCalendarConfigurationMBean Interface.....</b>	<b>209</b>
<b>com.bea.wli.management.configuration EmailEventGenChannelConfiguration Class.....</b>	<b>216</b>
<b>com.bea.wli.management.configuration EmailEventGenConfigurationMBean Interface.....</b>	<b>230</b>
<b>com.bea.wli.management.configuration EmailEventGeneratorsMBean Interface.....</b>	<b>233</b>
<b>com.bea.wli.management.configuration FileEventGenChannelConfiguration Class.....</b>	<b>237</b>
<b>com.bea.wli.management.configuration FileEventGenConfigurationMBean Interface.....</b>	<b>252</b>
<b>com.bea.wli.management.configuration FileEventGeneratorsMBean Interface.....</b>	<b>255</b>
<b>com.bea.wli.management.configuration GenericEventGenChannelConfiguration Class.....</b>	<b>259</b>
<b>com.bea.wli.management.configuration GenericEventGenConfigurationMBean Interface.....</b>	<b>263</b>

# Table of Contents

<b>com.bea.wli.management.configuration GenericEventGeneratorsMBean Interface.....</b>	<b>266</b>
<b>com.bea.wli.management.configuration JMSEventGenChannelConfiguration Class.....</b>	<b>270</b>
<b>com.bea.wli.management.configuration JMSEventGenConfigurationMBean Interface.....</b>	<b>275</b>
<b>com.bea.wli.management.configuration JMSEventGeneratorsMBean Interface.....</b>	<b>279</b>
<b>com.bea.wli.management.configuration MessageBrokerConfigurationMBean Interface.....</b>	<b>284</b>
<b>com.bea.wli.management.configuration NodeMethodInfo Class.....</b>	<b>288</b>
<b>com.bea.wli.management.configuration PasswordStoreConfigurationMBean Interface.....</b>	<b>292</b>
<b>com.bea.wli.management.configuration ProcessConfigurationMBean Interface.....</b>	<b>297</b>
<b>com.bea.wli.management.configuration ProcessControlMBean Interface.....</b>	<b>318</b>
<b>com.bea.wli.management.configuration ProcessNodeInfo Interface.....</b>	<b>323</b>
<b>com.bea.wli.management.configuration ProcessVariable Class.....</b>	<b>326</b>
<b>com.bea.wli.management.configuration SampleConfigurationMBean Interface.....</b>	<b>328</b>
<b>com.bea.wli.management.configuration ServiceBrokerControlMBean Interface.....</b>	<b>330</b>
<b>com.bea.wli.management.configuration TimerEventGenChannelConfiguration Class.....</b>	<b>335</b>
<b>com.bea.wli.management.configuration TimerEventGenConfigurationMBean Interface.....</b>	<b>343</b>
<b>com.bea.wli.management.configuration TimerEventGeneratorsMBean Interface.....</b>	<b>346</b>
<b>com.bea.wli.management.configuration TrackingLevel Class.....</b>	<b>350</b>
<b>com.bea.wli.management.configuration UserInfoConfigurationMBean Interface.....</b>	<b>354</b>
<b>com.bea.wli.management.configuration VersioningConfigurationMBean Interface.....</b>	<b>358</b>
<b>com.bea.wli.management.configuration WLIconsoleEventGenConfigurationMBean Interface.....</b>	<b>367</b>
<b>com.bea.wli.management.configuration WorklistConfigurationMBean Interface.....</b>	<b>370</b>
<b>com.bea.wli.management.runtime Package.....</b>	<b>373</b>
<b>com.bea.wli.management.runtime ArchiverRuntimeMBean Interface.....</b>	<b>375</b>

# Table of Contents

<b>com.bea.wli.management.runtime ChildInstance Interface.....</b>	<b>379</b>
<b>com.bea.wli.management.runtime EmailEventGenRuntimeMBean Interface.....</b>	<b>380</b>
<b>com.bea.wli.management.runtime FileEventGeneratorsRuntimeMBean Interface.....</b>	<b>384</b>
<b>com.bea.wli.management.runtime FileEventGenRuntimeMBean Interface.....</b>	<b>387</b>
<b>com.bea.wli.management.runtime GenericEventGenRuntimeMBean Interface.....</b>	<b>391</b>
<b>com.bea.wli.management.runtime HttpEventGenRuntimeMBean Interface.....</b>	<b>395</b>
<b>com.bea.wli.management.runtime JmsEventGenRuntimeMBean Interface.....</b>	<b>399</b>
<b>com.bea.wli.management.runtime MessageBrokerChannelValue Class.....</b>	<b>404</b>
<b>com.bea.wli.management.runtime MessageBrokerSubscriberValue Class.....</b>	<b>408</b>
<b>com.bea.wli.management.runtime MessageTrackingRuntimeException Class.....</b>	<b>411</b>
<b>com.bea.wli.management.runtime MessageTrackingRuntimeMBean Interface.....</b>	<b>413</b>
<b>com.bea.wli.management.runtime MsgBrokerRuntimeMBean Interface.....</b>	<b>420</b>
<b>com.bea.wli.management.runtime ParentInstance Interface.....</b>	<b>425</b>
<b>com.bea.wli.management.runtime ProcessInstanceInfo Interface.....</b>	<b>426</b>
<b>com.bea.wli.management.runtime ProcessInstanceQuery Class.....</b>	<b>430</b>
<b>com.bea.wli.management.runtime ProcessInstanceQueryResult Class.....</b>	<b>439</b>
<b>com.bea.wli.management.runtime ProcessRuntimeMBean Interface.....</b>	<b>442</b>
<b>com.bea.wli.management.runtime ProcessStatistics Interface.....</b>	<b>453</b>
<b>com.bea.wli.management.runtime ProcessVariableValue Interface.....</b>	<b>456</b>
<b>com.bea.wli.management.runtime SampleRuntimeMBean Interface.....</b>	<b>460</b>
<b>com.bea.wli.management.runtime TimerEventGeneratorsRuntimeMBean Interface.....</b>	<b>462</b>
<b>com.bea.wli.management.runtime TimerEventGenRuntimeMBean Interface.....</b>	<b>465</b>
<b>com.bea.wli.management.runtime WorklistRuntimeMBean Interface.....</b>	<b>469</b>

# Table of Contents

<b>com.bea.wli.management.runtime WorkSubstituteRuntimeMBean Interface.....</b>	<b>483</b>
<b>com.bea.wli.management.runtime WrappedThrowable Class.....</b>	<b>487</b>
<b>com.bea.wli.user.api Package.....</b>	<b>490</b>
<b>com.bea.wli.user.api UserInfo Interface.....</b>	<b>491</b>
<b>com.bea.wli.user.api UserInfoException Class.....</b>	<b>494</b>
<b>com.bea.wli.worklist.api Package.....</b>	<b>496</b>
<b>com.bea.wli.worklist.api ActionType Class.....</b>	<b>497</b>
<b>com.bea.wli.worklist.api Assignee Class.....</b>	<b>503</b>
<b>com.bea.wli.worklist.api AssignmentException Class.....</b>	<b>506</b>
<b>com.bea.wli.worklist.api BusinessTime Class.....</b>	<b>510</b>
<b>com.bea.wli.worklist.api ManagementException Class.....</b>	<b>512</b>
<b>com.bea.wli.worklist.api StateType Class.....</b>	<b>516</b>
<b>com.bea.wli.worklist.api TaskCreation Class.....</b>	<b>519</b>
<b>com.bea.wli.worklist.api TaskInfo Class.....</b>	<b>525</b>
<b>com.bea.wli.worklist.api TaskMessage Class.....</b>	<b>534</b>
<b>com.bea.wli.worklist.api TaskSelector Class.....</b>	<b>536</b>
<b>com.bea.wli.worklist.api WorklistManager Interface.....</b>	<b>556</b>
<b>com.bea.wli.worklist.api WorkSubstituteManager Interface.....</b>	<b>574</b>
<b>com.bea.wli.worklist.api WorkSubstituteRule Class.....</b>	<b>577</b>
<b>com.bea.wlxt Package.....</b>	<b>581</b>
<b>com.bea.wlxt XT Interface.....</b>	<b>582</b>
<b>com.bea.wlxt XTPool Interface.....</b>	<b>598</b>
<b>com.bea.xml Package.....</b>	<b>600</b>



## Table of Contents

<b>com.bea.xml XmlObjectList Interface.....</b>	<b>601</b>
<b>com.bea.xml XmlObjectList.Factory Class.....</b>	<b>605</b>
<b>com.bea.xml XmlUtils Class.....</b>	<b>607</b>
<b>com.bea.xml XmlUtilsException Class.....</b>	<b>609</b>

# Java Class Reference

This page provides links to the WebLogic Integration API Javadoc for WebLogic Integration controls, business processes (JPDs), data transformation, Worklist, business calendars, WLXT, XML, and so on.

## Topics Included in This Section

[com.bea.control Package](#)

[com.bea.data Package](#)

[com.bea.jpd Package](#)

[com.bea.transform Package](#)

[com.bea.wli.bpm.proxy Package](#)

[com.bea.wli.calendar.api Package](#)

[com.bea.wli.management.archiving Package](#)

[com.bea.wli.management.configuration Package](#)

[com.bea.wli.management.runtime Package](#)

[com.bea.wli.user.api Package](#)

[com.bea.wli.worklist.api Package](#)

[com.bea.wlxt Package](#)

[com.bea.xml Package](#)

# com.bea.control Package

## Interface Summary

B2BCallback	
B2BCallbackControl	
EBXMLControl	WebLogic Integration Control for ebXML protocol This is a base ebXML control that can be used to send and receive messages in the form of callback.
EBXMLControl.Callback	
EmailControl	Email control base interface.
FileControl	File Control base interface
HttpControl	HTTP Control base interface
MQControl	MQ Control base interface
ProcessControl	The process control is used to call a sub-process from a parent process.
PublishControl	
RosettaNetControl	Basic RosettaNet control.
RosettaNetControl.Callback	The following methods are available after a JCX instance is created.
ServiceBrokerControl	
SubscriptionControl	
SubscriptionControl.Callback	
TaskControl	The Task control enables a business process to create a single Task instance, manage its state and data, and provide callback methods that report status.
TaskControl.Callback	
TaskWorkerControl	The Task Worker control enables a business process or UI to assume ownership of Tasks, work on them, and complete them.
TPMControl	
WliJMSControl	
WliJMSControl.Callback	

## Exception Summary

ProcessControlException
TPMControlException

# B2BCallback Interface

*public interface* **B2BCallback**

extends `ExtendedServiceControl`

---

## *All Superinterfaces*

`Asynchronous`, `Control`, `ExtendedServiceControl`, `Serializable`,  
`ServiceControl`, `ServiceProxy`, `XMLControl`

## *All Known Subinterfaces*

`B2BCallbackControl`

---

## Nested Class Summary

### Nested classes from interface `com.bea.control.Control`

`Control.Callback`

### Nested classes from interface `com.bea.control.ServiceControl`

`ServiceControl.Callback`

## Method Summary

### Methods from interface `com.bea.wlw.runtime.core.control.ExtendedServiceControl`

`getEchoConversationID`, `getRecipientCertificate`, `getTargetService`,  
`setEchoConversationID`, `setRecipientCertificate`, `setTargetService`

### Methods from interface `com.bea.control.ServiceControl`

`getConversationID`, `getEndPoint`, `getInputHeaders`, `getPassword`,  
`getProtocol`, `getReliableMessageID`, `getUsername`, `reset`, `setClientCert`,  
`setConversationID`, `setEndPoint`, `setKeystore`, `setKeystore`,  
`setOutputHeaders`, `setPassword`, `setProtocol`, `setReliableMessageID`,  
`setUsername`, `useClientKeySSL`

# B2BCallbackControl Interface

*public interface B2BCallbackControl*

extends B2BCallback

---

## All Superinterfaces

Asynchronous, B2BCallback, Control, ExtendedServiceControl, Serializable, ServiceControl, ServiceProxy, XMLControl

---

## Nested Class Summary

### Nested classes from interface com.bea.control.Control

Control.Callback

### Nested classes from interface com.bea.control.ServiceControl

ServiceControl.Callback

## Method Summary

### Methods from interface com.bea.wlw.runtime.core.control.ExtendedServiceControl

getEchoConversationID, getRecipientCertificate, getTargetService, setEchoConversationID, setRecipientCertificate, setTargetService

### Methods from interface com.bea.control.ServiceControl

getConversationID, getEndPoint, getInputHeaders, getPassword, getProtocol, getReliableMessageID, getUsername, reset, setClientCert, setConversationID, setEndPoint, setKeystore, setKeystore, setOutputHeaders, setPassword, setProtocol, setReliableMessageID, setUsername, useClientKeySSL

# EBXMLControl Interface

*public interface* **EBXMLControl**

extends **Control**

WebLogic Integration Control for ebXML protocol This is a base ebXML control that can be used to send and receive messages in the form of callback. This control can be used to exchange messages for both ebXML 1.0 and ebXML 2.0 Messaging Service. The following methods are available after a JCX instance is created.

- **request()**

: Sends the specified message. Specified in the Control Send node in the initiator business process. Note: The default return type for the request method is void. However, you can also specify the return type to be XmlObject. If you use XmlObject as the return type, the content the XmlObject is the ebXML envelope data.

- **response()**

: Invoked when a business message is received. Specified in the Control Receive node in the initiator business process.

---

## All Superinterfaces

**Control**, **Serializable**

---

## Nested Class Summary

```
public
static EBXMLControl.Callback
interface
```

## Nested classes from interface com.bea.control.Control

**Control.Callback**

## Method Summary

```
public
EBXMLControlPropertiesDocument getProperties()
    Displays the current property settings.

public void
setProperty(EBXMLControlPropertiesDocument
xmlProps)
    Sets dynamic properties at run-time, such as the
    business ID of the sender or recipient.
```

## Method Detail

### getProperties() Method

```
public EBXMLControlPropertiesDocument getProperties()
```

Displays the current property settings.

#### Returns

*the dynamic properties for the EBXML control*

---

### setProperties(EBXMLControlPropertiesDocument) Method

```
public void setProperties(EBXMLControlPropertiesDocument xmlProps)
```

Sets dynamic properties at run-time, such as the business ID of the sender or recipient.

#### Parameters

*xmlProps*  
the dynamic properties for the EBXML control

# EBXMLControl.Callback Interface

*public static interface* **EBXMLControl.Callback**

*Enclosing interface*

EBXMLControl

---

## Method Summary

`public`  
    `void` **onAck**(*EnvelopeDocument* *ack*)  
        Invoked when an ebXML SOAP envelope containing an acknowledgement is received.

`public`  
    `void` **onError**(*EnvelopeDocument* *error*)  
        Invoked when an ebXML SOAP envelope containing an error is received.

## Method Detail

### onAck(EnvelopeDocument) Method

`public void` **onAck**(*EnvelopeDocument* *ack*)

Invoked when an ebXML SOAP envelope containing an acknowledgement is received. Base control method.

#### Parameters

*ack*  
    EBXML SOAP Envelope containing acknowledgement

---

### onError(EnvelopeDocument) Method

`public void` **onError**(*EnvelopeDocument* *error*)

Invoked when an ebXML SOAP envelope containing an error is received.

#### Parameters

*error*  
    EBXML SOAP Envelope containing Error



# EmailControl Interface

*public interface EmailControl*

extends Control

Email control base interface. The Email control sends e-mail to the specified list of recipients.

---

## All Superinterfaces

Control, Control, Serializable

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Field Summary

```
public
static ATTR_ATTACHMENTS
final String      String

public
static ATTR_BCC
final String      String

public
static ATTR_BODY
final String      String

public
static ATTR_CC
final String      String

public
static ATTR_CONTENTTYPE
final String      String

public
static ATTR_FROM_ADDRESS
final String      String

public
static ATTR_FROM_NAME
final String      String
```

```

    public
    static ATTR_HEADER_ENCODING
final String      String

    public
    static ATTR_PASSWORD
final String      String

    public
    static ATTR_PASSWORD_ALIAS
final String      String

    public
    static ATTR_REPLY_TO_ADDRESS
final String      String

    public
    static ATTR_REPLY_TO_NAME
final String      String

    public
    static ATTR_SMTP_ADDRESS
final String      String

    public
    static ATTR_SUBJECT
final String      String

    public
    static ATTR_TO
final String      String

    public
    static ATTR_USERNAME
final String      String

    public
    static TAG_EMAIL
final String      The email tag used to define email control attributes.

    public
    static TAG_SEND
final String      The @jbc:email tag is used to annotate a JBCX method that provides a convenient template
                    for sending email.

```

## Method Summary

```

    public
EmailControlPropertiesDocument getProperties()
    Get the properties set for this control.

    public void
setProperties(EmailControlPropertiesDocument
xmlProp)
    Sets the dynamic properties for the control.

```

## Field Detail

### ATTR\_ATTACHMENTS

```
public static final String ATTR_ATTACHMENTS
```

---

### ATTR\_BCC

```
public static final String ATTR_BCC
```

---

### ATTR\_BODY

```
public static final String ATTR_BODY
```

---

### ATTR\_CC

```
public static final String ATTR_CC
```

---

### ATTR\_CONTENTTYPE

```
public static final String ATTR_CONTENTTYPE
```

---

### ATTR\_FROM\_ADDRESS

```
public static final String ATTR_FROM_ADDRESS
```

---

### ATTR\_FROM\_NAME

```
public static final String ATTR_FROM_NAME
```

---

### ATTR\_HEADER\_ENCODING

```
public static final String ATTR_HEADER_ENCODING
```

---

### ATTR\_PASSWORD

```
public static final String ATTR_PASSWORD
```

---

## **ATTR\_PASSWORD\_ALIAS**

```
public static final String ATTR_PASSWORD_ALIAS
```

---

## **ATTR\_REPLY\_TO\_ADDRESS**

```
public static final String ATTR_REPLY_TO_ADDRESS
```

---

## **ATTR\_REPLY\_TO\_NAME**

```
public static final String ATTR_REPLY_TO_NAME
```

---

## **ATTR\_SMTP\_ADDRESS**

```
public static final String ATTR_SMTP_ADDRESS
```

---

## **ATTR\_SUBJECT**

```
public static final String ATTR_SUBJECT
```

---

## **ATTR\_TO**

```
public static final String ATTR_TO
```

---

## **ATTR\_USERNAME**

```
public static final String ATTR_USERNAME
```

---

## **TAG\_EMAIL**

```
public static final String TAG_EMAIL
```

The email tag used to define email control attributes.

---

## **TAG\_SEND**

```
public static final String TAG_SEND
```

The @jbc:email tag is used to annotate a JBCX method that provides a convenient template for sending email.

## Method Detail

### getProperties() Method

```
public EmailControlPropertiesDocument getProperties()
```

Get the properties set for this control.

#### Returns

*the dynamic properties for the control. NULL if the control has not been initialized yet.*

---

### setProperties(EmailControlPropertiesDocument) Method

```
public void setProperties(EmailControlPropertiesDocument xmlProp)
```

Sets the dynamic properties for the control.

#### Parameters

*xmlProp*  
the dynamic properties for the control.

com.bea.control

# FileControl Interface

*public interface FileControl*

extends Control

File Control base interface

---

## *All Superinterfaces*

Control, Control, Serializable

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Field Summary

```
public
static ATTR_ARCHIVE_DIR
final String      String

public
static ATTR_CREATE_MODE
final String      String

public
static ATTR_CREATE_MODE_OVERWRITE
final String      String

public
static ATTR_CREATE_MODE_RENAME
final String      String

public
static ATTR_CURRENT_DIR
final String      String

public
static ATTR_ENCODING
final String      String

public
static ATTR_ERROR_DIR
final String      String
```

```

    public
    static ATTR_FILE_CONTENT
final String      String

    public
    static ATTR_FILE_MASK
final String      String

    public
    static ATTR_FTP_HOST_NAME
final String      String

    public
    static ATTR_FTP_LOCAL_DIR
final String      String

    public
    static ATTR_FTP_USER_NAME
final String      String

    public
    static ATTR_IO_TYPE
final String      String

    public
    static ATTR_IO_TYPE_APPEND
final String      String

    public
    static ATTR_IO_TYPE_READ
final String      String

    public
    static ATTR_IO_TYPE_READLINE
final String      String

    public
    static ATTR_IO_TYPE_WRITE
final String      String

    public
    static ATTR_POST_READ
final String      String

    public
    static ATTR_POST_READ_ARCHIVE
final String      String

    public
    static ATTR_POST_READ_DELETE
final String      String

    public
    static ATTR_RECORD_SIZE
final String      String

    public
    static ATTR_SUFFIX_NAME
final String      String

    public
    static ATTR_SUFFIX_TYPE
final String      String

```

```

    public
    static ATTR_SUFFIX_TYPE_INDEX
final String      String

    public
    static ATTR_SUFFIX_TYPE_TIMESTAMP
final String      String

    public
    static DYNAMIC_PROPERTIES_TAG
final String      String

    public
    static TAG_FILE
final String      The file tag is used to define file control attributes

    public
    static TAG_FILE_OPERATION
final String      The @jc:file-operation tag is used to annotate a JBCX method that provides a convenient
                  template for the operations on files such as Read, Write and Append.

```

## Method Summary

```

    public void
        copy(String fileName)
        Copy the current file to the specified file name.

    public void
        delete()
        Delete the current file.

    public
FileControlFileListDocument getFiles()
        Returns Iterator for the File objects matching the
        specified file mask in the current directory.

    public
FileControlPropertiesDocument getProperties()
        Get the dynamic properties for the control

    public void
        rename(String fileName)
        Rename the current file to the specified file name.

    public void
        reset()
        Reset the file control by closing any in-progress
        operations such as: readLine, readRecord and append.

    public void
        setProperties(FileControlPropertiesDocument
xmlProp)
        Set the dynamic properties for the control

```



## Field Detail

### ATTR\_ARCHIVE\_DIR

```
public static final String ATTR_ARCHIVE_DIR
```

---

### ATTR\_CREATE\_MODE

```
public static final String ATTR_CREATE_MODE
```

---

### ATTR\_CREATE\_MODE\_OVERWRITE

```
public static final String ATTR_CREATE_MODE_OVERWRITE
```

---

### ATTR\_CREATE\_MODE\_RENAME

```
public static final String ATTR_CREATE_MODE_RENAME
```

---

### ATTR\_CURRENT\_DIR

```
public static final String ATTR_CURRENT_DIR
```

---

### ATTR\_ENCODING

```
public static final String ATTR_ENCODING
```

---

### ATTR\_ERROR\_DIR

```
public static final String ATTR_ERROR_DIR
```

---

### ATTR\_FILE\_CONTENT

```
public static final String ATTR_FILE_CONTENT
```

---

### ATTR\_FILE\_MASK

```
public static final String ATTR_FILE_MASK
```

---

**ATTR\_FTP\_HOST\_NAME**

```
public static final String ATTR_FTP_HOST_NAME
```

---

**ATTR\_FTP\_LOCAL\_DIR**

```
public static final String ATTR_FTP_LOCAL_DIR
```

---

**ATTR\_FTP\_USER\_NAME**

```
public static final String ATTR_FTP_USER_NAME
```

---

**ATTR\_IO\_TYPE**

```
public static final String ATTR_IO_TYPE
```

---

**ATTR\_IO\_TYPE\_APPEND**

```
public static final String ATTR_IO_TYPE_APPEND
```

---

**ATTR\_IO\_TYPE\_READ**

```
public static final String ATTR_IO_TYPE_READ
```

---

**ATTR\_IO\_TYPE\_READLINE**

```
public static final String ATTR_IO_TYPE_READLINE
```

---

**ATTR\_IO\_TYPE\_WRITE**

```
public static final String ATTR_IO_TYPE_WRITE
```

---

**ATTR\_POST\_READ**

```
public static final String ATTR_POST_READ
```

---

**ATTR\_POST\_READ\_ARCHIVE**

```
public static final String ATTR_POST_READ_ARCHIVE
```

---

## ATTR\_POST\_READ\_DELETE

```
public static final String ATTR_POST_READ_DELETE
```

---

## ATTR\_RECORD\_SIZE

```
public static final String ATTR_RECORD_SIZE
```

---

## ATTR\_SUFFIX\_NAME

```
public static final String ATTR_SUFFIX_NAME
```

---

## ATTR\_SUFFIX\_TYPE

```
public static final String ATTR_SUFFIX_TYPE
```

---

## ATTR\_SUFFIX\_TYPE\_INDEX

```
public static final String ATTR_SUFFIX_TYPE_INDEX
```

---

## ATTR\_SUFFIX\_TYPE\_TIMESTAMP

```
public static final String ATTR_SUFFIX_TYPE_TIMESTAMP
```

---

## DYNAMIC\_PROPERTIES\_TAG

```
public static final String DYNAMIC_PROPERTIES_TAG
```

---

## TAG\_FILE

```
public static final String TAG_FILE
```

The file tag is used to define file control attributes

---

## TAG\_FILE\_OPERATION

```
public static final String TAG_FILE_OPERATION
```

The @jc:file-operation tag is used to annotate a JBCX method that provides a convenient template for the operations on files such as Read, Write and Append.

## Method Detail

### copy(String) Method

```
public void copy(String fileName)
    throws FileNotFoundException, IOException, SecurityException, IllegalArgumentException
```

Copy the current file to the specified file name.

#### Parameters

*fileName*  
name of the file name to be copied to

#### Exceptions

*FileNotFoundException*  
if the file is not present  
*IOException*  
reading the file  
*SecurityException*  
reading the file  
*IllegalArgumentException*  
if invalid file name is specified

---

### delete() Method

```
public void delete()
    throws FileNotFoundException, IOException, SecurityException
```

Delete the current file.

#### Exceptions

*FileNotFoundException*  
if the file is not present  
*IOException*  
reading the file  
*SecurityException*  
reading the file

---

### getFiles() Method

```
public FileControlFileListDocument getFiles()
```

Returns Iterator for the File objects matching the specified file mask in the current directory.

**Returns**

*an iterator for the File objects for the files in the current directory matching the file mask*

---

**getProperties() Method**

```
public FileControlPropertiesDocument getProperties()
```

Get the dynamic properties for the control

**Returns**

*a FileControlPropertiesDocument object*

---

**rename(String) Method**

```
public void rename(String fileName)
    throws FileNotFoundException, IOException, SecurityException, IllegalArgumentException
```

Rename the current file to the specified file name.

**Parameters**

*fileName*  
name of the file to be renamed to

**Exceptions**

*FileNotFoundException*  
if the file is not present  
*IOException*  
reading the file  
*SecurityException*  
reading the file  
*IllegalArgumentException*  
if invalid file name is specified

---

**reset() Method**

```
public void reset()
```

Reset the file control by closing any in-progress operations such as: readLine, readRecord and append.

---

**setProperties(FileControlPropertiesDocument) Method**

```
public void setProperties(FileControlPropertiesDocument xmlProp)
    throws IllegalArgumentException
```

Set the dynamic properties for the control

### Parameters

*xmlProp*

the dynamic properties for the control

### Exceptions

*IllegalArgumentException*

if invalid properties are specified

# HttpControl Interface

*public interface* **HttpControl**

extends **Control**

HTTP Control base interface

---

## All Superinterfaces

**Control**, **Serializable**

---

## Nested Class Summary

Nested classes from interface **com.bea.control.Control**

**Control.Callback**

## Method Summary

```
public
com.bea.wli.control.httpCookies.CookiesDocument getCookies()
                                                    Get the cookies

public
com.bea.wli.control.httpHeaders.HeadersDocument getResponseHeaders()
                                                    Get the response headers

public void
                                                    setClientSideSSL(String keyStoreType,
                                                    keyStoreLocation, String keyStorePassword)
                                                    Sets the Client side SSL properties for

public void
                                                    setConnectionRetrycount(int retryCount)
                                                    Number of times the application will
                                                    specified it will be tried once

public void
                                                    setConnectionTimeout(int timeoutInMilliseconds)
                                                    Sets the timeout until a connection is

public void
                                                    setCookies(com.bea.wli.control.httpCookies cookies)
                                                    Set the Cookies
```

```
public void setPropertys(com.bea.wli.control.dyna
                        propsDoc)
    Sets the dynamic properties for the co
```

```
public void
    setProxyConfig(String host, int port,
                  String password)
    Sets the Proxy server properties for th
```

```
public void
    setServerSideSSL(String trustStoreL
                    hostVerificationFlag)
    Sets the Server side SSL properties fo
```

## Method Detail

### getCookies() Method

```
public com.bea.wli.control.httpCookies.CookiesDocument getCookies()
    throws Exception
```

Get the cookies

#### Exceptions

*Exception*

---

### getResponseHeaders() Method

```
public com.bea.wli.control.httpHeaders.HeadersDocument getResponseHeaders()
    throws Exception
```

Get the response headers

#### Exceptions

*Exception*

---

### setClientSideSSL(String, String, String, String) Method

```
public void setClientSideSSL(String keyStoreType,
                             String keyStoreLocation,
                             String keyStorePassword,
                             String keyPassword)
    throws Exception
```

Sets the Client side SSL properties for the control.



**Parameters***keyStoreType*

Key Store Type. There are two keystore types (JKS, PKCS12). If user doesn't give anything, the default key store type would be selected

*keyStoreLocation*

Key Store location needs to be set for two way ssl

*keyStorePassword*

Key store password needs to be set for two way ssl

*keyPassword*

Private key password needs to be set for two way ssl

**Exceptions***Exception*

---

**setConnectionRetrycount(int) Method**

```
public void setConnectionRetrycount(int retryCount)
    throws Exception
```

Number of times the application will retry for the request. If not specified it will be tried once

**Parameters***retryCount*

Numbers of retry request

**Exceptions***Exception*

---

**setConnectionTimeout(int) Method**

```
public void setConnectionTimeout(int timeoutInMilliseconds)
    throws Exception
```

Sets the timeout until a connection is established. A timeout value of zero means the timeout is not used. The default value is zero.

**Parameters***timeoutInMilliseconds*

Connection timeout in milli seconds

## Exceptions

*Exception*

---

### setCookies(com.bea.wli.control.httpCookies.CookiesDocument) Method

```
public void setCookies(com.bea.wli.control.httpCookies.CookiesDocument cookies)
    throws Exception
```

Set the Cookies

#### Parameters

*cookies*

the cookies to be set for the Http request

#### Exceptions

*Exception*

---

### setProperties(com.bea.wli.control.dynamicProperties.HttpControlPropertiesDocument) Method

```
public void setProperties(com.bea.wli.control.dynamicProperties.HttpControlPropertiesDocument p
    throws IllegalArgumentException
```

Sets the dynamic properties for the control.

#### Parameters

*propsDoc*

the dynamic properties for the control.

#### Exceptions

*IllegalArgumentException*

---

### setProxyConfig(String, int, String, String) Method

```
public void setProxyConfig(String host,
    int port,
    String userName,
    String password)
    throws IllegalArgumentException
```

Sets the Proxy server properties for the control.

## Parameters

*host*

proxy host name

*port*

proxy host port

*userName*

user Name for the connecting to proxy host

*password*

password for the user connecting to proxy host

## Exceptions

*IllegalArgumentException*

---

## setServerSideSSL(String, boolean) Method

```
public void setServerSideSSL(String trustStoreLocation,  
                             boolean hostVerificationFlag)  
    throws Exception
```

Sets the Server side SSL properties for the control.

## Parameters

*trustStoreLocation*

the trust store location needs to be specified for one way ssl

*hostVerificationFlag*

Host Verification Ignored flag.If true, then host name varification will be ignored

## Exceptions

*Exception*

com.bea.control

# MQControl Interface

*public interface MQControl*

extends Control

MQ Control base interface

---

## All Superinterfaces

Control, Serializable

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Method Summary

```
public void
    begin()
        Begins a MQ transaction

public void
    commit()
        Commits a MQ transaction

public byte[]
    getMessageAsBytes(String queueName,
com.bea.wli.control.mqmdHeaders.MQMDHeader,
        Gets a byte array(binary) message from
        the generic getMessage function

public String
    getMessageAsString(String queueName,
com.bea.wli.control.mqmdHeaders.MQMDHeader,
        Gets a String(text) message from
        generic getMessage function

public XmlObject
    getMessageAsXml(String queueName,
com.bea.wli.control.mqmdHeaders.MQMDHeader,
        Gets a XmlObject(xml) message from
        the generic getMessage function
```

```

        public putMessageAsBytes(byte[] message)
com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument
        Puts a byte array(binary) message into the generic putMessage function

        public
com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument putMessageAsString(String message)
com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument
        Puts a String(text) message into the generic putMessage function

        public
com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument putMessageAsXml(XmlObject message)
com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument
        Puts a XmlObject(xml) message into the generic putMessage function

        public void
rollback()
        Rolls back a MQ transaction

        public void
setDynamicProperties(com.bea.wli.control.mqDynPropsDoc mqDynPropsDoc)
        Sets the dynamic properties for the MQMDHeadersDocument
    
```

## Method Detail

### begin() Method

```

public void begin()
    throws ResourceException
    
```

Begins a MQ transaction

#### Exceptions

*ResourceException*  
if transaction state is invalid

---

### commit() Method

```

public void commit()
    throws ResourceException
    
```

Commits a MQ transaction

#### Exceptions

*ResourceException*  
if transaction state is invalid

---

## getMessageAsBytes(String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method

```
public byte[] getMessageAsBytes(String queue,
                                com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```

Gets a byte array(binary) message from the queue This function calls the generic getMessage function

### Parameters

*queue*

the queue from which the message is to be got

*mqmd*

the MQMDHeaders document containing the MQMD attributes based on which the message is to be got

### Returns

*byte[]* the byte array representing the message got

### Exceptions

*ResourceException*

if any exception occurs while get

---

## getMessageAsString(String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method

```
public String getMessageAsString(String queue,
                                com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```

Gets a String(text) message from the queue This function calls the generic getMessage function

### Parameters

*queue*

the queue from which the message is to be got

*mqmd*

the MQMDHeaders document containing the MQMD attributes based on which the message is to be got

### Returns

*String* the String representing the message got

**Exceptions***ResourceException*

if any exception occurs while get

**getMessageAsXml(String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method**

```
public XmlObject getMessageAsXml(String queue,
                                com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```

Gets a XmlObject(xml) message from the queue This function calls the generic getMessage function

**Parameters***queue*

the queue from which the message is to be got

*mqmd*

the MQMDHeaders document containing the MQMD attributes based on which the message is to be got

**Returns**

*XmlObject* the *XmlObject* representing the message got

**Exceptions***ResourceException*

if any exception occurs while get

**putMessageAsBytes(byte[], String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method**

```
public com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument putMessageAsBytes(byte[] message,
                                     String queue,
                                     com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```

Puts a byte array(binary) message into the queue This function calls the generic putMessage function

**Parameters***message*

the byte array message to be put into the queue

*queue*

the queue to which the message is to be put

*mqmd*

the MQMDHeaders document containing the MQMD attributes of the message to be put

## Returns

*MQMDHeadersDocument* representing the attributes of the message put

## Exceptions

*ResourceException*

if any exception occurs while put

---

## **putMessageAsString(String, String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method**

```
public com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument putMessageAsString(String message,
                                                                              String queue,
                                                                              com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```

Puts a String(text) message into the queue This function calls the generic putMessage function

## Parameters

*message*

the String message to be put into the queue

*queue*

the queue to which the message is to be put

*mqmd*

the MQMDHeaders document containing the MQMD attributes of the message to be put

## Returns

*MQMDHeadersDocument* representing the attributes of the message put

## Exceptions

*ResourceException*

if any exception occurs while put

---

## **putMessageAsXml(XmlObject, String, com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument) Method**

```
public com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument putMessageAsXml(XmlObject message,
                                                                              String queue,
                                                                              com.bea.wli.control.mqmdHeaders.MQMDHeadersDocument mqmd)
    throws ResourceException
```



Puts a XmlObject(xml) message into the queue This function calls the generic putMessage function

### Parameters

*message*

the xml message to be put into the queue

*queue*

the queue to which the message is to be put

*mqmd*

the MQMDHeaders document containing the MQMD attributes of the message to be put

### Returns

*MQMDHeadersDocument* representing the attributes of the message put

### Exceptions

*ResourceException*

if any exception occurs while put

---

## rollback() Method

```
public void rollback()  
    throws ResourceException
```

Rolls back a MQ transaction

### Exceptions

*ResourceException*

if transaction state is invalid

---

## setDynamicProperties(com.bea.wli.control.mqDynamicProperties.MQDynamicPropertiesDoc) Method

```
public void setDynamicProperties(com.bea.wli.control.mqDynamicProperties.MQDynamicPropertiesDoc
```

Sets the dynamic properties for the control.

### Parameters

*mqDynPropsDoc*

the MQDynamicProperties document containing the dynamic properties to be set

# ProcessControl Interface

*public interface* **ProcessControl**

extends Asynchronous, Control, ServiceProxy

The process control is used to call a sub-process from a parent process. Sub-process callbacks are routed to the caller process. The sub-process must be in the same domain as the caller. The invocation is dispatched internally over RMI without going thru http. If the target process is versioned, the actual version invoked depends on the version strategy of the caller (as specified by the jpd:version annotation). The process control runtime properties can be configured externally through dynamic properties and xquery selectors.

---

## All Superinterfaces

Asynchronous, Control, Serializable, ServiceProxy

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Method Summary

```

        public String
                                getConversationID()
                                Returns the conversation ID currently associated
                                with this control or null.

        public
ProcessControlPropertiesDocument getProperties()
                                Returns the current control properties.

        public URI
                                getTargetURI()
                                Returns the target URI or null if not assigned.

        public String
                                getUsername()
                                Gets credential information.

        public void
                                reset()
                                Resets the conversational state of the proxy; this
                                could result in dropping an existing conversation.

```

## Java Class Reference

```
public void setConversationID(String conversationID)
    Manual control over conversation ID

public void
    setPassword(String password)
    Sets credential information.

public void
    setProperties(ProcessControlPropertiesDocument
    props)
    Sets the dynamic properties for the control

public void
    setTargetURI(URI uri)
    Manual control over the target uri.

public void
    setUsername(String username)
    Sets credential information.
```

## Method Detail

### getConversationID() Method

```
public String getConversationID()
```

Returns the conversation ID currently associated with this control or null.

#### Returns

*conversation id*

---

### getProperties() Method

```
public ProcessControlPropertiesDocument getProperties()
```

Returns the current control properties. Note that this does not include username/password.

#### Returns

*the properties*

---

### getTargetURI() Method

```
public URI getTargetURI()
```

Returns the target URI or null if not assigned.

#### Returns

*target uri*

## getUsername() Method

```
public String getUsername()
```

Gets credential information.

### Returns

*username*

---

## reset() Method

```
public void reset()
```

Resets the conversational state of the proxy; this could result in dropping an existing conversation.

---

## setConversationID(String) Method

```
public void setConversationID(String conversationID)
```

Manual control over conversation ID

---

## setPassword(String) Method

```
public void setPassword(String password)
```

Sets credential information.

---

## setProperties(ProcessControlPropertiesDocument) Method

```
public void setProperties(ProcessControlPropertiesDocument props)
    throws ProcessControlException
```

Sets the dynamic properties for the control

### Parameters

*props*  
the dynamic properties for the control

### Exceptions

*ProcessControlException*

---

## setTargetURI(URI) Method

```
public void setTargetURI(URI uri)
    throws ProcessControlException
```

Manual control over the target uri.

### Exceptions

*ProcessControlException*

---

## setUsername(String) Method

```
public void setUsername(String username)
```

Sets credential information.

# ProcessControlException Class

*public class ProcessControlException*

extends `ControlException`

---

## *Hierarchy*

Object  
  Throwable  
    Exception  
      RuntimeException  
      ControlException  
      ProcessControlException

## *All Implemented Interfaces*

Serializable

---

## Constructor Summary

*ProcessControlException(String message, Throwable cause)*

*ProcessControlException(String message)*

## Method Summary

Methods from `com.bea.control.ControlException`

`getNestedException`

Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### ProcessControlException

```
public ProcessControlException(String message,  
                               Throwable cause)
```

---

### ProcessControlException

```
public ProcessControlException(String message)
```

# PublishControl Interface

*public interface **PublishControl***

extends `Control`

---

## *All Superinterfaces*

`Control`, `Control`, `Serializable`

---

## Nested Class Summary

Nested classes from interface `com.bea.control.Control`

`Control.Callback`

## Method Summary

```
public
PublishControlPropertiesDocument getProperties()
    Sets the dynamic properties for the control

public void
    setOutputHeaders(Element[] headers)
    Temporarily sets the message headers to use in the
    next publish operation

public void
    setProperties(PublishControlPropertiesDocument
    props)
    Sets the dynamic properties for the control
```

## Method Detail

### **getProperties()** Method

```
public PublishControlPropertiesDocument getProperties()
```

Sets the dynamic properties for the control

### Returns

*the current properties for the control*

---



## **setOutputHeaders(Element[]) Method**

```
public void setOutputHeaders(Element[] headers)
```

Temporarily sets the message headers to use in the next publish operation

### **Parameters**

*headers*

headers to set

---

## **setProperties(PublishControlPropertiesDocument) Method**

```
public void setProperties(PublishControlPropertiesDocument props)
```

Sets the dynamic properties for the control

### **Parameters**

*props*

the dynamic properties for the control

# RosettaNetControl Interface

*public interface* **RosettaNetControl**

extends **Control**

Basic RosettaNet control. Send/Receive functionality is left to the extensible control, for customization. The following methods are available after a JCX instance is created.

- **sendMessage()**

: Sends the specified message to a trading partner.

---

## All Superinterfaces

**Control**, **Serializable**

---

## Nested Class Summary

```

    public static interface RosettaNetControl.Callback
    {
        The following methods are available after a JCX instance is created.
    }

```

## Nested classes from interface com.bea.control.Control

**Control.Callback**

## Method Summary

```

    public String
        getProcessTimeout()
        Returns the amount of time to wait for the entire
        business process to complete before timing out.

    public
RosettaNetControlPropertiesDocument getProperties()
        Get the dynamic properties for the RosettaNet
        control

    public int
        getRetryCount()
        Returns the maximum number of retries for
        sending the message

    public String
        getRetryInterval()
        Returns the time interval between retries

```

## Java Class Reference

`public void sendAck()`

Sends an acknowledgement of receipt of a business message to a trading partner and accepts the request.

`public void`

`sendError()`

Sends an Exception

`public void`

`sendReject()`

Sends an acknowledgement of receipt of the business message to a trading partner, but rejects the request.

`public void`

`setProperties(RosettaNetControlPropertiesDoc  
xmlPropsDoc)`

Sets dynamic properties at run–time, such as the business ID of the sender or recipient.

## Method Detail

### getProcessTimeout() Method

`public String getProcessTimeout()`

Returns the amount of time to wait for the entire business process to complete before timing out.

---

### getProperties() Method

`public RosettaNetControlPropertiesDocument getProperties()`

Get the dynamic properties for the RosettaNet control

#### Returns

*the dynamic properties for the RosettaNet control*

---

### getRetryCount() Method

`public int getRetryCount()`

Returns the maximum number of retries for sending the message

---

### getRetryInterval() Method

`public String getRetryInterval()`

Returns the time interval between retries

**sendAck() Method**

```
public void sendAck()
```

Sends an acknowledgement of receipt of a business message to a trading partner and accepts the request.

---

**sendError() Method**

```
public void sendError()
```

Sends an Exception

---

**sendReject() Method**

```
public void sendReject()
```

Sends an acknowledgement of receipt of the business message to a trading partner, but rejects the request.

---

**setProperties(RosettaNetControlPropertiesDocument) Method**

```
public void setProperties(RosettaNetControlPropertiesDocument xmlPropsDoc)
```

Sets dynamic properties at run-time, such as the business ID of the sender or recipient. Use this method to set the debug mode during testing.

**Parameters**

*xmlPropsDoc*  
the dynamic properties for the RosettaNet control

## RosettaNetControl.Callback Interface

*public static interface RosettaNetControl.Callback*

The following methods are available after a JCX instance is created.

- onAck()

: Invoked when a RosettaNet acknowledgement is received.

- onError()

: Invoked when a RosettaNet error notification is received.

- onMessage()

: Invoked when a RosettaNet business message is received.

*Enclosing interface*

RosettaNetControl

---

# ServiceBrokerControl Interface

*public interface ServiceBrokerControl*

extends ServiceControl

---

## All Superinterfaces

Asynchronous, Control, Serializable, ServiceControl, ServiceProxy, XMLControl

---

## Nested Class Summary

### Nested classes from interface com.bea.control.Control

Control.Callback

### Nested classes from interface com.bea.control.ServiceControl

ServiceControl.Callback

## Method Summary

```
public
ServiceBrokerControlPropertiesDocument getProperties()
    Returns the current control properties.

public void
    setProperties(ServiceBrokerControlPropertiesDocument props)
    This method sets the control properties dynamically.
```

### Methods from interface com.bea.control.ServiceControl

getConversationID, getEndPoint, getInputHeaders, getPassword, getProtocol, getReliableMessageID, getUsername, reset, setClientCert, setConversationID, setEndPoint, setKeystore, setKeystore, setOutputHeaders, setPassword, setProtocol, setReliableMessageID, setUsername, useClientKeySSL

## Method Detail

### getProperties() Method

```
public ServiceBrokerControlPropertiesDocument getProperties()
```

Returns the current control properties. Note that this includes all properties except security-related settings like username/password, keyAlias/keyPassword and keyStoreLocation/keyStorePassword.

#### Returns

*the properties*

---

### setProperties(ServiceBrokerControlPropertiesDocument) Method

```
public void setProperties(ServiceBrokerControlPropertiesDocument props)
    throws Exception
```

This method sets the control properties dynamically. Properties set by this method over-ride the static annotations.

#### Parameters

*props*  
control properties

#### Exceptions

*Exception*

# SubscriptionControl Interface

*public interface SubscriptionControl*

extends `Control`

---

## All Superinterfaces

`Control`, `Control`, `Serializable`

---

## Nested Class Summary

```
public
static SubscriptionControl.Callback
interface
```

Nested classes from interface `com.bea.control.Control`

`Control.Callback`

## Method Summary

```
public
void subscribe()
    Subscribes the control to the message broker.

public
void unsubscribe()
    Unsubscribes the control from the message broker, stopping further events (messages) from
    being delivered to the control.
```

## Method Detail

### **subscribe()** Method

```
public void subscribe()
```

Subscribes the control to the message broker. If the subscription uses a filter expression, then the default filter value will be used. If no default filter value is defined in the annotations, then a `null` filter value will be used, meaning that any filter result will trigger a callback.

---



## **unsubscribe() Method**

```
public void unsubscribe()
```

Unsubscribes the control from the message broker, stopping further events (messages) from being delivered to the control.

com.bea.control

## SubscriptionControl.Callback Interface

*public static interface SubscriptionControl.Callback*

*Enclosing interface*

SubscriptionControl

---

# TaskControl Interface

*public interface TaskControl*

extends Control

The Task control enables a business process to create a single Task instance, manage its state and data, and provide callback methods that report status. Each Task control operates on a single active Task instance. You can extend this control.

---

## All Superinterfaces

Control, Control, Serializable

---

## Nested Class Summary

```
public
static TaskControl.Callback
interface
```

Nested classes from interface com.bea.control.Control

Control.Callback

## Field Summary

```
public
static ATTR_ASSIGN_ALGORITHM
final String String

public
static ATTR_ASSIGN_GROUP
final String String

public
static ATTR_ASSIGN_USER
final String String

public
static ATTR_CREATE_CAN_BE_ABORTED
final String String

public
static ATTR_CREATE_CAN_BE_REASSIGNED
final String String

public
static
```

```

final String ATTR_CREATE_CAN_BE_RETURNED
        String

        public
        static ATTR_CREATE_CLAIM_CALENDAR
final String
        String

        public
        static ATTR_CREATE_CLAIM_DUE_BUSINESS_DATE
final String
        String

        public
        static ATTR_CREATE_CLAIM_DUE_DATE
final String
        String

        public
        static ATTR_CREATE_CLAIM_USER_CALENDAR
final String
        String

        public
        static ATTR_CREATE_COMMENT
final String
        String

        public
        static ATTR_CREATE_COMPLETION_CALENDAR
final String
        String

        public
        static ATTR_CREATE_COMPLETION_DUE_BUSINESS_DATE
final String
        String

        public
        static ATTR_CREATE_COMPLETION_DUE_DATE
final String
        String

        public
        static ATTR_CREATE_COMPLETION_USER_CALENDAR
final String
        String

        public
        static ATTR_CREATE_DESCRIPTION
final String
        String

        public
        static ATTR_CREATE_NAME
final String
        String

        public
        static ATTR_CREATE_OWNER
final String
        String

        public
        static ATTR_CREATE_PRIORITY
final String
        String

        public
        static ATTR_CREATE_REQUEST
final String
        String

        public
        static ATTR_CREATE_REQUEST_MIME_TYPE
final String
        String

```

```

    public
    static ATTR_GET_PROPERTY_NAME
final String      String

    public
    static ATTR_REMOVE_PROPERTY_NAME
final String      String

    public
    static ATTR_SET_PROPERTY_MAP
final String      String

    public
    static ATTR_SET_PROPERTY_NAME
final String      String

    public
    static ATTR_SET_PROPERTY_VALUE
final String      String

    public
    static ATTR_UPDATE_CAN_BE_ABORTED
final String      String

    public
    static ATTR_UPDATE_CAN_BE_REASSIGNED
final String      String

    public
    static ATTR_UPDATE_CAN_BE_RETURNED
final String      String

    public
    static ATTR_UPDATE_CLAIM_CALENDAR
final String      String

    public
    static ATTR_UPDATE_CLAIM_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_UPDATE_CLAIM_DUE_DATE
final String      String

    public
    static ATTR_UPDATE_CLAIM_USER_CALENDAR
final String      String

    public
    static ATTR_UPDATE_COMMENT
final String      String

    public
    static ATTR_UPDATE_COMPLETION_CALENDAR
final String      String

    public
    static ATTR_UPDATE_COMPLETION_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_UPDATE_COMPLETION_DUE_DATE
final String      String

```

```

    public
    static ATTR_UPDATE_COMPLETION_USER_CALENDAR
final String      String

    public
    static ATTR_UPDATE_OWNER
final String      String

    public
    static ATTR_UPDATE_PRIORITY
final String      String

    public
    static ATTR_UPDATE_REQUEST
final String      String

    public
    static ATTR_UPDATE_REQUEST_MIME_TYPE
final String      String

    public
    static ATTR_UPDATE_RESPONSE
final String      String

    public
    static ATTR_UPDATE_RESPONSE_MIME_TYPE
final String      String

    public
    static CALENDAR_TYPE_NAME
final Integer     Integer

    public
    static CALENDAR_TYPE_SYSTEM
final Integer     Integer

    public
    static CALENDAR_TYPE_USER_OR_GROUP
final Integer     Integer

    public
    static TAG_ADVANCED
final String      the control advanced options jc tag.

    public
    static TAG_ASSIGNEE
final String      the control assignement jc tag.

    public
    static TAG_TASK
final String      the control jc tag.

    public
    static TAG_TASK_ABORT
final String      String

    public
    static TAG_TASK_ASSIGN
final String      The @jc:create tag is used to annotate a JBCX method that assign or reassign a task.

    public
    static TAG_TASK_CREATE
final String      The @jc:create tag is used to annotate a JBCX method that creates a new task.

```

```

    public
    static TAG_TASK_GET_PROPERTY
final String      The return type of the method JBCX is used to determine what is returned.

    public
    static TAG_TASK_GET_REQUEST
final String      The return type of the method JBCX is used to determine what is returned.

    public
    static TAG_TASK_GET_RESPONSE
final String      The return type of the method JBCX is used to determine what is returned.

    public
    static TAG_TASK_REMOVE_PROPERTY
final String      String

    public
    static TAG_TASK_RESUME
final String      String

    public
    static TAG_TASK_SET_PROPERTY
final String      String

    public
    static TAG_TASK_SUSPEND
final String      String

    public
    static TAG_TASK_UPDATE
final String      The @jc:task-update tag is used to annotate a JBCX method that update the task
                    properties.

```

## Method Summary

```

    public String
        createTask(TaskCreationXMLDocument doc)
                    Create and optionally assigns a new task using an XML
                    interface

    public String
        getTaskId()

    public TaskInfo
        getTaskInfo()

    public
TaskInfoXMLDocument getTaskInfoXMLDocument()

    public String[]
        getTaskPropertiesNames()

    public void
        setTaskId(String taskId)
                    Set the TaskId the control is linked to.

    public void
        updateTask(TaskUpdateXMLDocument doc)

```

## Field Detail

### ATTR\_ASSIGN\_ALGORITHM

```
public static final String ATTR_ASSIGN_ALGORITHM
```

---

### ATTR\_ASSIGN\_GROUP

```
public static final String ATTR_ASSIGN_GROUP
```

---

### ATTR\_ASSIGN\_USER

```
public static final String ATTR_ASSIGN_USER
```

---

### ATTR\_CREATE\_CAN\_BE\_ABORTED

```
public static final String ATTR_CREATE_CAN_BE_ABORTED
```

---

### ATTR\_CREATE\_CAN\_BE\_REASSIGNED

```
public static final String ATTR_CREATE_CAN_BE_REASSIGNED
```

---

### ATTR\_CREATE\_CAN\_BE\_RETURNED

```
public static final String ATTR_CREATE_CAN_BE_RETURNED
```

---

### ATTR\_CREATE\_CLAIM\_CALENDAR

```
public static final String ATTR_CREATE_CLAIM_CALENDAR
```

---

### ATTR\_CREATE\_CLAIM\_DUE\_BUSINESS\_DATE

```
public static final String ATTR_CREATE_CLAIM_DUE_BUSINESS_DATE
```

---

### ATTR\_CREATE\_CLAIM\_DUE\_DATE

```
public static final String ATTR_CREATE_CLAIM_DUE_DATE
```

---



## **ATTR\_CREATE\_CLAIM\_USER\_CALENDAR**

```
public static final String ATTR_CREATE_CLAIM_USER_CALENDAR
```

---

## **ATTR\_CREATE\_COMMENT**

```
public static final String ATTR_CREATE_COMMENT
```

---

## **ATTR\_CREATE\_COMPLETION\_CALENDAR**

```
public static final String ATTR_CREATE_COMPLETION_CALENDAR
```

---

## **ATTR\_CREATE\_COMPLETION\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_CREATE_COMPLETION_DUE_BUSINESS_DATE
```

---

## **ATTR\_CREATE\_COMPLETION\_DUE\_DATE**

```
public static final String ATTR_CREATE_COMPLETION_DUE_DATE
```

---

## **ATTR\_CREATE\_COMPLETION\_USER\_CALENDAR**

```
public static final String ATTR_CREATE_COMPLETION_USER_CALENDAR
```

---

## **ATTR\_CREATE\_DESCRIPTION**

```
public static final String ATTR_CREATE_DESCRIPTION
```

---

## **ATTR\_CREATE\_NAME**

```
public static final String ATTR_CREATE_NAME
```

---

## **ATTR\_CREATE\_OWNER**

```
public static final String ATTR_CREATE_OWNER
```

---

## **ATTR\_CREATE\_PRIORITY**

```
public static final String ATTR_CREATE_PRIORITY
```

---

**ATTR\_CREATE\_REQUEST**

```
public static final String ATTR_CREATE_REQUEST
```

---

**ATTR\_CREATE\_REQUEST\_MIME\_TYPE**

```
public static final String ATTR_CREATE_REQUEST_MIME_TYPE
```

---

**ATTR\_GET\_PROPERTY\_NAME**

```
public static final String ATTR_GET_PROPERTY_NAME
```

---

**ATTR\_REMOVE\_PROPERTY\_NAME**

```
public static final String ATTR_REMOVE_PROPERTY_NAME
```

---

**ATTR\_SET\_PROPERTY\_MAP**

```
public static final String ATTR_SET_PROPERTY_MAP
```

---

**ATTR\_SET\_PROPERTY\_NAME**

```
public static final String ATTR_SET_PROPERTY_NAME
```

---

**ATTR\_SET\_PROPERTY\_VALUE**

```
public static final String ATTR_SET_PROPERTY_VALUE
```

---

**ATTR\_UPDATE\_CAN\_BE\_ABORTED**

```
public static final String ATTR_UPDATE_CAN_BE_ABORTED
```

---

**ATTR\_UPDATE\_CAN\_BE\_REASSIGNED**

```
public static final String ATTR_UPDATE_CAN_BE_REASSIGNED
```

---

**ATTR\_UPDATE\_CAN\_BE\_RETURNED**

```
public static final String ATTR_UPDATE_CAN_BE_RETURNED
```

---

## **ATTR\_UPDATE\_CLAIM\_CALENDAR**

```
public static final String ATTR_UPDATE_CLAIM_CALENDAR
```

---

## **ATTR\_UPDATE\_CLAIM\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_UPDATE_CLAIM_DUE_BUSINESS_DATE
```

---

## **ATTR\_UPDATE\_CLAIM\_DUE\_DATE**

```
public static final String ATTR_UPDATE_CLAIM_DUE_DATE
```

---

## **ATTR\_UPDATE\_CLAIM\_USER\_CALENDAR**

```
public static final String ATTR_UPDATE_CLAIM_USER_CALENDAR
```

---

## **ATTR\_UPDATE\_COMMENT**

```
public static final String ATTR_UPDATE_COMMENT
```

---

## **ATTR\_UPDATE\_COMPLETION\_CALENDAR**

```
public static final String ATTR_UPDATE_COMPLETION_CALENDAR
```

---

## **ATTR\_UPDATE\_COMPLETION\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_UPDATE_COMPLETION_DUE_BUSINESS_DATE
```

---

## **ATTR\_UPDATE\_COMPLETION\_DUE\_DATE**

```
public static final String ATTR_UPDATE_COMPLETION_DUE_DATE
```

---

## **ATTR\_UPDATE\_COMPLETION\_USER\_CALENDAR**

```
public static final String ATTR_UPDATE_COMPLETION_USER_CALENDAR
```

---

## **ATTR\_UPDATE\_OWNER**

```
public static final String ATTR_UPDATE_OWNER
```

---

**ATTR\_UPDATE\_PRIORITY**

```
public static final String ATTR_UPDATE_PRIORITY
```

---

**ATTR\_UPDATE\_REQUEST**

```
public static final String ATTR_UPDATE_REQUEST
```

---

**ATTR\_UPDATE\_REQUEST\_MIME\_TYPE**

```
public static final String ATTR_UPDATE_REQUEST_MIME_TYPE
```

---

**ATTR\_UPDATE\_RESPONSE**

```
public static final String ATTR_UPDATE_RESPONSE
```

---

**ATTR\_UPDATE\_RESPONSE\_MIME\_TYPE**

```
public static final String ATTR_UPDATE_RESPONSE_MIME_TYPE
```

---

**CALENDAR\_TYPE\_NAME**

```
public static final Integer CALENDAR_TYPE_NAME
```

---

**CALENDAR\_TYPE\_SYSTEM**

```
public static final Integer CALENDAR_TYPE_SYSTEM
```

---

**CALENDAR\_TYPE\_USER\_OR\_GROUP**

```
public static final Integer CALENDAR_TYPE_USER_OR_GROUP
```

---

**TAG\_ADVANCED**

```
public static final String TAG_ADVANCED
```

the control advanced options jc tag.

---

## TAG\_ASSIGNEE

```
public static final String TAG_ASSIGNEE
```

the control assignement jc tag.

---

## TAG\_TASK

```
public static final String TAG_TASK
```

the control jc tag.

---

## TAG\_TASK\_ABORT

```
public static final String TAG_TASK_ABORT
```

---

## TAG\_TASK\_ASSIGN

```
public static final String TAG_TASK_ASSIGN
```

The @jc:create tag is used to annotate a JBCX method that assign or reassign a task. The algorithm name is used to determine the way a task is assigned. If no name is pecified the default algorithm is used and the args must be of type String[]; In any case the attribute "args" is required and must be of type Object[].

---

## TAG\_TASK\_CREATE

```
public static final String TAG_TASK_CREATE
```

The @jc:create tag is used to annotate a JBCX method that creates a new task. This method can at most be called once by control and returns the taskId. The callbacks are registered to the new task created

---

## TAG\_TASK\_GET\_PROPERTY

```
public static final String TAG_TASK_GET_PROPERTY
```

The return type of the method JBCX is used to determine what is returned.

---

## TAG\_TASK\_GET\_REQUEST

```
public static final String TAG_TASK_GET_REQUEST
```

The return type of the method JBCX is used to determine what is returned.

## TAG\_TASK\_GET\_RESPONSE

```
public static final String TAG_TASK_GET_RESPONSE
```

The return type of the method JBCX is used to determine what is returned.

---

## TAG\_TASK\_REMOVE\_PROPERTY

```
public static final String TAG_TASK_REMOVE_PROPERTY
```

---

## TAG\_TASK\_RESUME

```
public static final String TAG_TASK_RESUME
```

---

## TAG\_TASK\_SET\_PROPERTY

```
public static final String TAG_TASK_SET_PROPERTY
```

---

## TAG\_TASK\_SUSPEND

```
public static final String TAG_TASK_SUSPEND
```

---

## TAG\_TASK\_UPDATE

```
public static final String TAG_TASK_UPDATE
```

The @jc:task-update tag is used to annotate a JBCX method that update the task properties. You can modify one or more properties at the same time.

## Method Detail

### createTask(TaskCreationXMLDocument) Method

```
public String createTask(TaskCreationXMLDocument doc)
```

Create and optionally assigns a new task using an XML interface

**Parameters***doc*

the task parameters

**Returns***the taskId*

---

**getTaskId() Method**

```
public String getTaskId()
```

---

**getTaskInfo() Method**

```
public TaskInfo getTaskInfo()
```

---

**getTaskInfoXMLDocument() Method**

```
public TaskInfoXMLDocument getTaskInfoXMLDocument()
```

---

**getTaskPropertiesNames() Method**

```
public String[] getTaskPropertiesNames()
```

---

**setTaskId(String) Method**

```
public void setTaskId(String taskId)
```

Set the TaskId the control is linked to. The taskId can only be set once. When a task is created, its taskId is set to the control if no taskId has been set before or return an exception otherwise. If you want to work with more than one taskId you can use multiple control or the WorklistControl.

---

**updateTask(TaskUpdateXMLDocument) Method**

```
public void updateTask(TaskUpdateXMLDocument doc)
```

# TaskControl.Callback Interface

*public static interface TaskControl.Callback*

*Enclosing interface*

TaskControl

---

## Field

### Summary

```
public
static ATTR_TASK_EVENT_RESPONSE
final String String

public
static ATTR_TASK_EVENT_TIME
final String String

public
static ATTR_TASK_EVENT_TYPE
final String String

public
static ATTR_TASK_EVENT_USER
final String String

public
static TAG_TASK_EVENT
final String Internal callback method used to implement user-defined callbacks.
```

## Field Detail

### ATTR\_TASK\_EVENT\_RESPONSE

```
public static final String ATTR_TASK_EVENT_RESPONSE
```

---

### ATTR\_TASK\_EVENT\_TIME

```
public static final String ATTR_TASK_EVENT_TIME
```

---

### ATTR\_TASK\_EVENT\_TYPE

```
public static final String ATTR_TASK_EVENT_TYPE
```

---



## ATTR\_TASK\_EVENT\_USER

```
public static final String ATTR_TASK_EVENT_USER
```

---

## TAG\_TASK\_EVENT

```
public static final String TAG_TASK_EVENT
```

Internal callback method used to implement user-defined callbacks. When a task is set to the control instance, all the callback are registered to the task.

# TaskWorkerControl Interface

*public interface TaskWorkerControl*

extends Control

The Task Worker control enables a business process or UI to assume ownership of Tasks, work on them, and complete them. It offers administrative operations, such as operations to start, stop, delete, and assign. The Task Worker controls allow operations on several Task instances simultaneously. You can extend this control.

---

## All Superinterfaces

Control, Control, Serializable

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Field Summary

```

    public
    static ATTR_ASSIGN_ALGORITHM
final String      String

    public
    static ATTR_ASSIGN_GROUP
final String      String

    public
    static ATTR_ASSIGN_USER
final String      String

    public
    static ATTR_CLAIM_CLAIMANT
final String      String

    public
    static ATTR_CREATE_CAN_BE_ABORTED
final String      String

    public
    static ATTR_CREATE_CAN_BE_REASSIGNED
final String      String

```

```

    public ATTR_CREATE_CAN_BE_RETURNED
    static      String
final String

    public
    static ATTR_CREATE_CLAIM_CALENDAR
final String      String

    public
    static ATTR_CREATE_CLAIM_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_CREATE_CLAIM_DUE_DATE
final String      String

    public
    static ATTR_CREATE_CLAIM_USER_CALENDAR
final String      String

    public
    static ATTR_CREATE_COMMENT
final String      String

    public
    static ATTR_CREATE_COMPLETION_CALENDAR
final String      String

    public
    static ATTR_CREATE_COMPLETION_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_CREATE_COMPLETION_DUE_DATE
final String      String

    public
    static ATTR_CREATE_COMPLETION_USER_CALENDAR
final String      String

    public
    static ATTR_CREATE_DESCRIPTION
final String      String

    public
    static ATTR_CREATE_NAME
final String      String

    public
    static ATTR_CREATE_OWNER
final String      String

    public
    static ATTR_CREATE_PRIORITY
final String      String

    public
    static ATTR_CREATE_REQUEST
final String      String

    public
    static ATTR_CREATE_REQUEST_MIME_TYPE
final String      String

```

```

    public
    static ATTR_GET_PROPERTY_NAME
final String      String

    public
    static ATTR_REMOVE_PROPERTY_NAME
final String      String

    public
    static ATTR_SELECT_ASSIGNED_GROUP
final String      String

    public
    static ATTR_SELECT_ASSIGNED_USER
final String      String

    public
    static ATTR_SELECT_CLAIM_DUE_DATE_AFTER
final String      String

    public
    static ATTR_SELECT_CLAIM_DUE_DATE_BEFORE
final String      String

    public
    static ATTR_SELECT_CLAIMANT
final String      String

    public
    static ATTR_SELECT_COMPLETION_DUE_DATE_AFTER
final String      String

    public
    static ATTR_SELECT_COMPLETION_DUE_DATE_BEFORE
final String      String

    public
    static ATTR_SELECT_CREATION_DATE_AFTER
final String      String

    public
    static ATTR_SELECT_CREATION_DATE_BEFORE
final String      String

    public
    static ATTR_SELECT_MAX_PRIORITY
final String      String

    public
    static ATTR_SELECT_MIN_PRIORITY
final String      String

    public
    static ATTR_SELECT_PROPERTY_NAME
final String      String

    public
    static ATTR_SELECT_PROPERTY_VALUE
final String      String

    public
    static ATTR_SELECT_SELECTOR
final String      String

```

```

    public
    static ATTR_SELECT_STATES
final String      String

    public
    static ATTR_SELECT_TASK_COMMENT
final String      String

    public
    static ATTR_SELECT_TASK_ID
final String      String

    public
    static ATTR_SELECT_TASK_NAME
final String      String

    public
    static ATTR_SELECT_TASK_OWNER
final String      String

    public
    static ATTR_SET_PROPERTY_NAME
final String      String

    public
    static ATTR_SET_PROPERTY_VALUE
final String      String

    public
    static ATTR_UPDATE_CAN_BE_ABORTED
final String      String

    public
    static ATTR_UPDATE_CAN_BE_REASSIGNED
final String      String

    public
    static ATTR_UPDATE_CAN_BE_RETURNED
final String      String

    public
    static ATTR_UPDATE_CLAIM_CALENDAR
final String      String

    public
    static ATTR_UPDATE_CLAIM_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_UPDATE_CLAIM_DUE_DATE
final String      String

    public
    static ATTR_UPDATE_CLAIM_USER_CALENDAR
final String      String

    public
    static ATTR_UPDATE_COMMENT
final String      String

    public
    static ATTR_UPDATE_COMPLETION_CALENDAR
final String      String

```

```

    public
    static ATTR_UPDATE_COMPLETION_DUE_BUSINESS_DATE
final String      String

    public
    static ATTR_UPDATE_COMPLETION_DUE_DATE
final String      String

    public
    static ATTR_UPDATE_COMPLETION_USER_CALENDAR
final String      String

    public
    static ATTR_UPDATE_OWNER
final String      String

    public
    static ATTR_UPDATE_PRIORITY
final String      String

    public
    static ATTR_UPDATE_REQUEST
final String      String

    public
    static ATTR_UPDATE_REQUEST_MIME_TYPE
final String      String

    public
    static ATTR_UPDATE_RESPONSE
final String      String

    public
    static ATTR_UPDATE_RESPONSE_MIME_TYPE
final String      String

    public
    static TAG_TASK_WORKER
final String      the control jc tag.

    public
    static TAG_TASK_WORKER_ABORT
final String      String

    public
    static TAG_TASK_WORKER_ASSIGN
final String      The @jc:task-assign tag is used to annotate a JBCX method that assign or reassign a task.

    public
    static TAG_TASK_WORKER_CLAIM
final String      String

    public
    static TAG_TASK_WORKER_COMPLETE
final String      String

    public
    static TAG_TASK_WORKER_CREATE
final String      The @jc:create tag is used to annotate a JBCX method that creates a new task.

    public
    static TAG_TASK_WORKER_DELETE
final String      String

```

```

    public TAG_TASK_WORKER_GET_INFO
    static      The return type of the method JBCX is used to determine what is returned: TaskInfo[]
final String    TaskInfoXML[]

    public
    static TAG_TASK_WORKER_GET_PROPERTY
final String    String

    public
    static TAG_TASK_WORKER_GET_PROPERTY_NAME
final String    String

    public
    static TAG_TASK_WORKER_GET_REQUEST
final String    The return type of the method JBCX is used to determine what is returned.

    public
    static TAG_TASK_WORKER_GET_RESPONSE
final String    The return type of the method JBCX is used to determine what is returned.

    public
    static TAG_TASK_WORKER_REMOVE_PROPERTY
final String    String

    public
    static TAG_TASK_WORKER_RESUME
final String    String

    public
    static TAG_TASK_WORKER_RETURN
final String    String

    public
    static TAG_TASK_WORKER_SELECT
final String    The @jc:task-select tag is used in correlation with the other method tags of this control.

    public
    static TAG_TASK_WORKER_SET_PROPERTY
final String    String

    public
    static TAG_TASK_WORKER_START
final String    String

    public
    static TAG_TASK_WORKER_STOP
final String    String

    public
    static TAG_TASK_WORKER_SUSPEND
final String    String

    public
    static TAG_TASK_WORKER_UPDATE
final String    The @jc:task-update tag is used to annotate a JBCX method that update the task properties.

```

## Method Summary

```
public
    void archiveTasks()
        Archive the tasks that are in the state aborted or completed

public
    String createTask(TaskCreationXMLDocument doc)
        Create and optionally assigns a new task using an XML interface

public
    void purgeTasks()
        Purge the tasks that are ready to be purged: completed or aborted time > purgDelay.
```

## Field Detail

### ATTR\_ASSIGN\_ALGORITHM

```
public static final String ATTR_ASSIGN_ALGORITHM
```

---

### ATTR\_ASSIGN\_GROUP

```
public static final String ATTR_ASSIGN_GROUP
```

---

### ATTR\_ASSIGN\_USER

```
public static final String ATTR_ASSIGN_USER
```

---

### ATTR\_CLAIM\_CLAIMANT

```
public static final String ATTR_CLAIM_CLAIMANT
```

---

### ATTR\_CREATE\_CAN\_BE\_ABORTED

```
public static final String ATTR_CREATE_CAN_BE_ABORTED
```

---

### ATTR\_CREATE\_CAN\_BE\_REASSIGNED

```
public static final String ATTR_CREATE_CAN_BE_REASSIGNED
```

---

### ATTR\_CREATE\_CAN\_BE\_RETURNED

```
public static final String ATTR_CREATE_CAN_BE_RETURNED
```

---



## **ATTR\_CREATE\_CLAIM\_CALENDAR**

```
public static final String ATTR_CREATE_CLAIM_CALENDAR
```

---

## **ATTR\_CREATE\_CLAIM\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_CREATE_CLAIM_DUE_BUSINESS_DATE
```

---

## **ATTR\_CREATE\_CLAIM\_DUE\_DATE**

```
public static final String ATTR_CREATE_CLAIM_DUE_DATE
```

---

## **ATTR\_CREATE\_CLAIM\_USER\_CALENDAR**

```
public static final String ATTR_CREATE_CLAIM_USER_CALENDAR
```

---

## **ATTR\_CREATE\_COMMENT**

```
public static final String ATTR_CREATE_COMMENT
```

---

## **ATTR\_CREATE\_COMPLETION\_CALENDAR**

```
public static final String ATTR_CREATE_COMPLETION_CALENDAR
```

---

## **ATTR\_CREATE\_COMPLETION\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_CREATE_COMPLETION_DUE_BUSINESS_DATE
```

---

## **ATTR\_CREATE\_COMPLETION\_DUE\_DATE**

```
public static final String ATTR_CREATE_COMPLETION_DUE_DATE
```

---

## **ATTR\_CREATE\_COMPLETION\_USER\_CALENDAR**

```
public static final String ATTR_CREATE_COMPLETION_USER_CALENDAR
```

---

## **ATTR\_CREATE\_DESCRIPTION**

```
public static final String ATTR_CREATE_DESCRIPTION
```

---

**ATTR\_CREATE\_NAME**

```
public static final String ATTR_CREATE_NAME
```

---

**ATTR\_CREATE\_OWNER**

```
public static final String ATTR_CREATE_OWNER
```

---

**ATTR\_CREATE\_PRIORITY**

```
public static final String ATTR_CREATE_PRIORITY
```

---

**ATTR\_CREATE\_REQUEST**

```
public static final String ATTR_CREATE_REQUEST
```

---

**ATTR\_CREATE\_REQUEST\_MIME\_TYPE**

```
public static final String ATTR_CREATE_REQUEST_MIME_TYPE
```

---

**ATTR\_GET\_PROPERTY\_NAME**

```
public static final String ATTR_GET_PROPERTY_NAME
```

---

**ATTR\_REMOVE\_PROPERTY\_NAME**

```
public static final String ATTR_REMOVE_PROPERTY_NAME
```

---

**ATTR\_SELECT\_ASSIGNED\_GROUP**

```
public static final String ATTR_SELECT_ASSIGNED_GROUP
```

---

**ATTR\_SELECT\_ASSIGNED\_USER**

```
public static final String ATTR_SELECT_ASSIGNED_USER
```

---

**ATTR\_SELECT\_CLAIM\_DUE\_DATE\_AFTER**

```
public static final String ATTR_SELECT_CLAIM_DUE_DATE_AFTER
```

---

## **ATTR\_SELECT\_CLAIM\_DUE\_DATE\_BEFORE**

```
public static final String ATTR_SELECT_CLAIM_DUE_DATE_BEFORE
```

---

## **ATTR\_SELECT\_CLAIMANT**

```
public static final String ATTR_SELECT_CLAIMANT
```

---

## **ATTR\_SELECT\_COMPLETION\_DUE\_DATE\_AFTER**

```
public static final String ATTR_SELECT_COMPLETION_DUE_DATE_AFTER
```

---

## **ATTR\_SELECT\_COMPLETION\_DUE\_DATE\_BEFORE**

```
public static final String ATTR_SELECT_COMPLETION_DUE_DATE_BEFORE
```

---

## **ATTR\_SELECT\_CREATION\_DATE\_AFTER**

```
public static final String ATTR_SELECT_CREATION_DATE_AFTER
```

---

## **ATTR\_SELECT\_CREATION\_DATE\_BEFORE**

```
public static final String ATTR_SELECT_CREATION_DATE_BEFORE
```

---

## **ATTR\_SELECT\_MAX\_PRIORITY**

```
public static final String ATTR_SELECT_MAX_PRIORITY
```

---

## **ATTR\_SELECT\_MIN\_PRIORITY**

```
public static final String ATTR_SELECT_MIN_PRIORITY
```

---

## **ATTR\_SELECT\_PROPERTY\_NAME**

```
public static final String ATTR_SELECT_PROPERTY_NAME
```

---

## **ATTR\_SELECT\_PROPERTY\_VALUE**

```
public static final String ATTR_SELECT_PROPERTY_VALUE
```

---

**ATTR\_SELECT\_SELECTOR**

```
public static final String ATTR_SELECT_SELECTOR
```

---

**ATTR\_SELECT\_STATES**

```
public static final String ATTR_SELECT_STATES
```

---

**ATTR\_SELECT\_TASK\_COMMENT**

```
public static final String ATTR_SELECT_TASK_COMMENT
```

---

**ATTR\_SELECT\_TASK\_ID**

```
public static final String ATTR_SELECT_TASK_ID
```

---

**ATTR\_SELECT\_TASK\_NAME**

```
public static final String ATTR_SELECT_TASK_NAME
```

---

**ATTR\_SELECT\_TASK\_OWNER**

```
public static final String ATTR_SELECT_TASK_OWNER
```

---

**ATTR\_SET\_PROPERTY\_NAME**

```
public static final String ATTR_SET_PROPERTY_NAME
```

---

**ATTR\_SET\_PROPERTY\_VALUE**

```
public static final String ATTR_SET_PROPERTY_VALUE
```

---

**ATTR\_UPDATE\_CAN\_BE\_ABORTED**

```
public static final String ATTR_UPDATE_CAN_BE_ABORTED
```

---

**ATTR\_UPDATE\_CAN\_BE\_REASSIGNED**

```
public static final String ATTR_UPDATE_CAN_BE_REASSIGNED
```

---

## **ATTR\_UPDATE\_CAN\_BE\_RETURNED**

```
public static final String ATTR_UPDATE_CAN_BE_RETURNED
```

---

## **ATTR\_UPDATE\_CLAIM\_CALENDAR**

```
public static final String ATTR_UPDATE_CLAIM_CALENDAR
```

---

## **ATTR\_UPDATE\_CLAIM\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_UPDATE_CLAIM_DUE_BUSINESS_DATE
```

---

## **ATTR\_UPDATE\_CLAIM\_DUE\_DATE**

```
public static final String ATTR_UPDATE_CLAIM_DUE_DATE
```

---

## **ATTR\_UPDATE\_CLAIM\_USER\_CALENDAR**

```
public static final String ATTR_UPDATE_CLAIM_USER_CALENDAR
```

---

## **ATTR\_UPDATE\_COMMENT**

```
public static final String ATTR_UPDATE_COMMENT
```

---

## **ATTR\_UPDATE\_COMPLETION\_CALENDAR**

```
public static final String ATTR_UPDATE_COMPLETION_CALENDAR
```

---

## **ATTR\_UPDATE\_COMPLETION\_DUE\_BUSINESS\_DATE**

```
public static final String ATTR_UPDATE_COMPLETION_DUE_BUSINESS_DATE
```

---

## **ATTR\_UPDATE\_COMPLETION\_DUE\_DATE**

```
public static final String ATTR_UPDATE_COMPLETION_DUE_DATE
```

---

## **ATTR\_UPDATE\_COMPLETION\_USER\_CALENDAR**

```
public static final String ATTR_UPDATE_COMPLETION_USER_CALENDAR
```

---

**ATTR\_UPDATE\_OWNER**

```
public static final String ATTR_UPDATE_OWNER
```

---

**ATTR\_UPDATE\_PRIORITY**

```
public static final String ATTR_UPDATE_PRIORITY
```

---

**ATTR\_UPDATE\_REQUEST**

```
public static final String ATTR_UPDATE_REQUEST
```

---

**ATTR\_UPDATE\_REQUEST\_MIME\_TYPE**

```
public static final String ATTR_UPDATE_REQUEST_MIME_TYPE
```

---

**ATTR\_UPDATE\_RESPONSE**

```
public static final String ATTR_UPDATE_RESPONSE
```

---

**ATTR\_UPDATE\_RESPONSE\_MIME\_TYPE**

```
public static final String ATTR_UPDATE_RESPONSE_MIME_TYPE
```

---

**TAG\_TASK\_WORKER**

```
public static final String TAG_TASK_WORKER
```

the control jc tag.

---

**TAG\_TASK\_WORKER\_ABORT**

```
public static final String TAG_TASK_WORKER_ABORT
```

---

**TAG\_TASK\_WORKER\_ASSIGN**

```
public static final String TAG_TASK_WORKER_ASSIGN
```

The @jc:task-assign tag is used to annotate a JBCX method that assign or reassign a task. The algorithm name is used to determine the way a task is assigned. If no name is pecified the default algorithm is used and the args must be of type String[]; In any case the attribute "args" is required and must be of type Object[].

## **TAG\_TASK\_WORKER\_CLAIM**

```
public static final String TAG_TASK_WORKER_CLAIM
```

---

## **TAG\_TASK\_WORKER\_COMPLETE**

```
public static final String TAG_TASK_WORKER_COMPLETE
```

---

## **TAG\_TASK\_WORKER\_CREATE**

```
public static final String TAG_TASK_WORKER_CREATE
```

The @jc:create tag is used to annotate a JBCX method that creates a new task. This method returns the taskId of the created task.

---

## **TAG\_TASK\_WORKER\_DELETE**

```
public static final String TAG_TASK_WORKER_DELETE
```

---

## **TAG\_TASK\_WORKER\_GET\_INFO**

```
public static final String TAG_TASK_WORKER_GET_INFO
```

The return type of the method JBCX is used to determine what is returned: TaskInfo[]  
TaskInfoXML[]

---

## **TAG\_TASK\_WORKER\_GET\_PROPERTY**

```
public static final String TAG_TASK_WORKER_GET_PROPERTY
```

---

## **TAG\_TASK\_WORKER\_GET\_PROPERTY\_NAME**

```
public static final String TAG_TASK_WORKER_GET_PROPERTY_NAME
```

---

## **TAG\_TASK\_WORKER\_GET\_REQUEST**

```
public static final String TAG_TASK_WORKER_GET_REQUEST
```

The return type of the method JBCX is used to determine what is returned.

---

## **TAG\_TASK\_WORKER\_GET\_RESPONSE**

```
public static final String TAG_TASK_WORKER_GET_RESPONSE
```

The return type of the method JBCX is used to determine what is returned.

---

## **TAG\_TASK\_WORKER\_REMOVE\_PROPERTY**

```
public static final String TAG_TASK_WORKER_REMOVE_PROPERTY
```

---

## **TAG\_TASK\_WORKER\_RESUME**

```
public static final String TAG_TASK_WORKER_RESUME
```

---

## **TAG\_TASK\_WORKER\_RETURN**

```
public static final String TAG_TASK_WORKER_RETURN
```

---

## **TAG\_TASK\_WORKER\_SELECT**

```
public static final String TAG_TASK_WORKER_SELECT
```

The @jc:task-select tag is used in correlation with the other method tags of this control. It is used to determine the tasks concerned by the operation.

---

## **TAG\_TASK\_WORKER\_SET\_PROPERTY**

```
public static final String TAG_TASK_WORKER_SET_PROPERTY
```

---

## **TAG\_TASK\_WORKER\_START**

```
public static final String TAG_TASK_WORKER_START
```

---

## **TAG\_TASK\_WORKER\_STOP**

```
public static final String TAG_TASK_WORKER_STOP
```

---

## **TAG\_TASK\_WORKER\_SUSPEND**

```
public static final String TAG_TASK_WORKER_SUSPEND
```

---



## TAG\_TASK\_WORKER\_UPDATE

```
public static final String TAG_TASK_WORKER_UPDATE
```

The @jc:task-update tag is used to annotate a JBCX method that update the task properties. You can modify one or more properties at the same time.

## Method Detail

### archiveTasks() Method

```
public void archiveTasks()
```

Archive the tasks that are in the state aborted or completed

---

### createTask(TaskCreationXMLDocument) Method

```
public String createTask(TaskCreationXMLDocument doc)
```

Create and optionally assigns a new task using an XML interface

#### Parameters

*doc*

the task parameters

#### Returns

*the taskId*

---

### purgeTasks() Method

```
public void purgeTasks()
```

Purge the tasks that are ready to be purged: completed or aborted time > purgDelay. if archiver is on the task must be archived first.

# TPMControl Interface

*public interface TPMControl*

extends Control

---

## All Superinterfaces

Control, Control, Serializable

---

## Nested Class Summary

Nested classes from interface com.bea.control.Control

Control.Callback

## Method Summary

```
public
TradingPartnerDocument getBasicProperties(String partnerId)
    Gets the basic properties of a Trading Partner.

public EbxmlBindingDocument
getDefaultEbxmlBinding(String partnerId, String
    ebxmlVersion)
    Gets the default ebXML binding for a Trading Partner.

public
TradingPartnerDocument getDefaultPartner()
    Gets the default Trading Partner.

public
RosettanetBindingDocument getDefaultRosettanetBinding(String partnerId,
    String rnVersion)
    Gets the default Rosettanet binding for a Trading Partner.

public
ExtendedPropertySetDocument getExtendedPropertySet(String partnerId, String
    propertySetName)
    Gets an extended property set of a Trading Partner.

public
TradingPartnerDocument getPartnerById(String partnerId)
    Gets a Trading Partner.

public
TradingPartnerDocument getPartnerByName(String partnerName)
```

## Java Class Reference

Gets a Trading Partner.

```
public ServiceDocument
```

```
    getService(String serviceName)
```

Gets a Service.

```
    public
```

```
ServiceProfileDocument getServiceProfile(String serviceName, String
```

```
toPartnerId, String fromPartnerId)
```

Gets a ServiceProfile.

```
public EbxmlBindingDocument
```

```
    getServiceProfileEbxmlBinding(String serviceName,
```

```
String toPartnerId, String fromPartnerId, boolean
```

```
toPartnerBinding)
```

Gets the EbxmlBinding of a ServiceProfile.

```
    public
```

```
RosettanetBindingDocument getServiceProfileRosettanetBinding(String
```

```
serviceName, String toPartnerId, String fromPartnerId,
```

```
boolean toPartnerBinding)
```

Gets the RosettanetBinding of a ServiceProfile.

```
    public
```

```
WebServiceBindingDocument getServiceProfileWebServiceBinding(String
```

```
serviceName, String toPartnerId, String fromPartnerId,
```

```
boolean toPartnerBinding)
```

Gets the WebServiceBinding of a ServiceProfile.

## Method Detail

### getBasicProperties(String) Method

```
public TradingPartnerDocument getBasicProperties(String partnerId)
```

```
    throws TPMControlException
```

Gets the basic properties of a Trading Partner.

#### Parameters

*partnerId*

Business Id of the Trading Partner

#### Returns

*TradingPartnerDocument* XBean encapsulating the attributes of the TradingPartner schema in TPM

#### Exceptions

*TPMControlException*

---

## getDefaultEbxmlBinding(String, String) Method

```
public EbxmlBindingDocument getDefaultEbxmlBinding(String partnerId,  
                                                    String ebxmlVersion)  
    throws TPMControlException
```

Gets the default ebXML binding for a Trading Partner.

### Parameters

*partnerId*  
Business Id of the Trading Partner

*ebxmlVersion*  
ebXML version of the binding

### Returns

*EbxmlBindingDocument XBean encapsulating the XML fragment conforming to the EbxmlBinding schema in TPM*

### Exceptions

*TPMControlException*

---

## getDefaultPartner() Method

```
public TradingPartnerDocument getDefaultPartner()  
    throws TPMControlException
```

Gets the default Trading Partner.

### Returns

*TradingPartnerDocument XBean encapsulating the XML fragment conforming to the TradingPartner schema in TPM*

### Exceptions

*TPMControlException*

---

## getDefaultRosettanetBinding(String, String) Method

```
public RosettanetBindingDocument getDefaultRosettanetBinding(String partnerId,  
                                                              String rnVersion)  
    throws TPMControlException
```

Gets the default Rosettanet binding for a Trading Partner.

**Parameters**

*partnerId*  
Business Id of the Trading Partner

*rnVersion*  
Rosettanet version of the binding

**Returns**

*RosettanetBindingDocument XBean encapsulating the XML fragment conforming to the RosettanetBinding schema in TPM*

**Exceptions**

*TPMControlException*

---

**getExtendedPropertySet(String, String) Method**

```
public ExtendedPropertySetDocument getExtendedPropertySet(String partnerId,
                                                         String propertySetName)
    throws TPMControlException
```

Gets an extended property set of a Trading Partner.

**Parameters**

*partnerId*  
Business Id of the Trading Partner

*propertySetName*  
Name of the property set

**Returns**

*ExtendedPropertySetDocument XBean encapsulating the XML fragment conforming to the ExtendedPropertySet schema in TPM*

**Exceptions**

*TPMControlException*

---

**getPartnerById(String) Method**

```
public TradingPartnerDocument getPartnerById(String partnerId)
    throws TPMControlException
```

Gets a Trading Partner.

**Parameters**

*partnerId*  
Business Id of the Trading Partner

**Returns**

*TradingPartnerDocument XBean encapsulating the XML fragment conforming to the TradingPartner schema in TPM*

**Exceptions**

*TPMControlException*

---

**getPartnerByName(String) Method**

```
public TradingPartnerDocument getPartnerByName(String partnerName)
    throws TPMControlException
```

Gets a Trading Partner.

**Parameters**

*partnerName*  
Name of the Trading Partner

**Returns**

*TradingPartnerDocument XBean encapsulating the XML fragment conforming to the TradingPartner schema in TPM*

**Exceptions**

*TPMControlException*

---

**getService(String) Method**

```
public ServiceDocument getService(String serviceName)
    throws TPMControlException
```

Gets a Service.

**Parameters**

*serviceName*  
Name of the Service

**Returns**

*ServiceDocument XBean encapsulating the XML fragment conforming to the Service schema in TPM*

**Exceptions**

*TPMControlException*

---

**getServiceProfile(String, String, String) Method**

```
public ServiceProfileDocument getServiceProfile(String serviceName,
                                              String toPartnerId,
                                              String fromPartnerId)
    throws TPMControlException
```

Gets a ServiceProfile.

**Parameters**

*serviceName*  
Name of the Service

*toPartnerId*  
Business Id of the receiving Trading Partner

*fromPartnerId*  
Business Id of the sending Trading Partner

**Returns**

*ServiceProfileDocument XBean encapsulating the XML fragment conforming to the ServiceProfile schema in TPM*

**Exceptions**

*TPMControlException*

---

**getServiceProfileEbxmlBinding(String, String, String, boolean) Method**

```
public EbxmlBindingDocument getServiceProfileEbxmlBinding(String serviceName,
                                                         String toPartnerId,
                                                         String fromPartnerId,
                                                         boolean toPartnerBinding)
    throws TPMControlException
```

Gets the EbxmlBinding of a ServiceProfile.

**Parameters***serviceName*

Name of the Service

*toPartnerId*

Business Id of the receiving Trading Partner

*fromPartnerId*

Business Id of the sending Trading Partner

*toPartnerBinding*

'true' if requesting receiving Trading Partner Binding

**Returns***EbxmlBindingDocument* XBean encapsulating the XML fragment conforming to the *EbxmlBinding* schema in *TPM***Exceptions***TPMControlException***getServiceProfileRosettanetBinding(String, String, String, boolean) Method**

```

public RosettanetBindingDocument getServiceProfileRosettanetBinding(String serviceName,
                                                                    String toPartnerId,
                                                                    String fromPartnerId,
                                                                    boolean toPartnerBinding)

    throws TPMControlException

```

Gets the RosettanetBinding of a ServiceProfile.

**Parameters***serviceName*

Name of the Service

*toPartnerId*

Business Id of the receiving Trading Partner

*fromPartnerId*

Business Id of the sending Trading Partner

*toPartnerBinding*

'true' if requesting receiving Trading Partner Binding



**Returns**

*RosettanetBindingDocument XBean encapsulating the XML fragment conforming to the RosettanetBinding schema in TPM*

**Exceptions**

*TPMControlException*

---

**getServiceProfileWebServiceBinding(String, String, String, boolean) Method**

```
public WebServiceBindingDocument getServiceProfileWebServiceBinding(String serviceName,
                                                                    String toPartnerId,
                                                                    String fromPartnerId,
                                                                    boolean toPartnerBinding)
    throws TPMControlException
```

Gets the WebServiceBinding of a ServiceProfile.

**Parameters**

*serviceName*

Name of the Service

*toPartnerId*

Business Id of the receiving Trading Partner

*fromPartnerId*

Business Id of the sending Trading Partner

*toPartnerBinding*

'true' if requesting receiving Trading Partner Binding

**Returns**

*WebServiceBindingDocument XBean encapsulating the XML fragment conforming to the WebServiceBinding schema in TPM*

**Exceptions**

*TPMControlException*

# TPMControlException Class

*public class TPMControlException*

extends `ControlException`

---

## *Hierarchy*

```
Object
  Throwable
    Exception
      RuntimeException
        ControlException
          ControlException
            TPMControlException
```

## *All Implemented Interfaces*

`Serializable`

---

## Constructor Summary

*TPMControlException(String s, Throwable throwable)*

*TPMControlException(String s)*

## Method Summary

Methods from `com.bea.control.ControlException`

`getNestedException`

Methods from `java.lang.Throwable`

`fillInStackTrace, getCause, getLocalizedMessage, getMessage, getStackTrace, initCause, printStackTrace, printStackTrace, printStackTrace, setStackTrace, toString`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### TPMControlException

```
public TPMControlException(String s,  
                           Throwable throwable)
```

---

### TPMControlException

```
public TPMControlException(String s)
```

# WliJMSControl Interface

*public interface WliJMSControl*

extends JMSControl

---

## All Superinterfaces

Control, JMSControl, Serializable, XMLControl

---

## Nested Class Summary

```
public
static WliJMSControl.Callback
interface
```

### Nested classes from interface com.bea.control.Control

Control.Callback

### Nested classes from interface com.bea.control.JMSControl

JMSControl.Callback

## Field Summary

### Fields from interface com.bea.control.JMSControl

HEADER\_CORRELATIONID, HEADER\_DELIVERYMODE, HEADER\_EXPIRATION,  
HEADER\_MESSAGEID, HEADER\_PRIORITY, HEADER\_REDELIVERED, HEADER\_TIMESTAMP,  
HEADER\_TYPE

## Method Summary

```
public
JMSControlPropertiesDocument getControlProperties()
    This method returns the WliJMSControl's properties.

public void
    setProperties(JMSControlPropertiesDocument
jmsProps)
    This method sets the control properties dynamically.
```

**Methods from interface `com.bea.control.JMSControl`**

`getHeaders`, `getProperties`, `getSession`, `setHeaders`, `setProperties`,  
`subscribe`, `unsubscribe`

## Method Detail

### **`getControlProperties()` Method**

```
public JMSControlPropertiesDocument getControlProperties()
```

This method returns the `WliJMSControl`'s properties.

#### **Returns**

*JMSControlPropertiesDocument WliJMSControl dynamic properties*

---

### **`setProperties(JMSControlPropertiesDocument)` Method**

```
public void setProperties(JMSControlPropertiesDocument jmsProps)  
    throws ControlException
```

This method sets the control properties dynamically. Properties set by this method over-ride the static annotations.

#### **Parameters**

*jmsProps*  
control properties

#### **Exceptions**

*ControlException*

com.bea.control

## WliJMSSControl.Callback Interface

*public static interface WliJMSSControl.Callback*

extends JMSSControl.Callback

---

### *All Superinterfaces*

JMSSControl.Callback

### *Enclosing interface*

WliJMSSControl

---

# com.bea.data Package

## Interface Summary

**MessageAttachment** Represents part of an attachment in an ebXML or RosettaNet business message.

**RawData** This defines a Java interface for a potentially large piece of data (incoming message stored in the repository) which can be a process variable.

## Class Summary

**MessageAttachment.Factory** Static factory class for creating new instances of **MessageAttachment**

**RawData.Factory** Static factory class for creating new instances of **RawData**

## Exception Summary

**RawDataRuntimeException** Used to mark runtime exceptions for **RawData**.

# MessageAttachment Interface

*public interface MessageAttachment*

extends Serializable

Represents part of an attachment in an ebXML or RosettaNet business message. Provides methods for retrieving untyped XML or non-XML data from an attachment.

---

## All Superinterfaces

Serializable

---

## Nested Class Summary

```

    public static final class MessageAttachment.Factory
        Static factory class for creating new instances of MessageAttachment

```

## Field Summary

```

    public static final long serialVersionUID
        long

```

## Method Summary

```

    public RawData getRawData()
        Retrieves this portion of the attachment as non-XML (raw) data.

    public XmlObject getXmlObject()
        Retrieves this portion of the attachment as an XmlObject.

    public boolean isRawData()
        Returns true if this portion of the attachment contains non-XML (raw) data.

    public boolean isXmlObject()
        Returns true if this portion of the attachment contains untyped XML data.

    public String toString()
        Converts this portion of the attachment to a String.

```



## Field Detail

### serialVersionUID

```
public static final long serialVersionUID
```

## Method Detail

### getRawData() Method

```
public RawData getRawData()
```

Retrieves this portion of the attachment as non-XML (raw) data. Call `isRawData()` first to verify that the data is raw data.

---

### getXmlObject() Method

```
public XmlObject getXmlObject()
```

Retrieves this portion of the attachment as an `XmlObject`. Call `isXmlObject()` first to verify that the data is untyped XML.

---

### isRawData() Method

```
public boolean isRawData()
```

Returns true if this portion of the attachment contains non-XML (raw) data.

---

### isXmlObject() Method

```
public boolean isXmlObject()
```

Returns true if this portion of the attachment contains untyped XML data.

---

### toString() Method

```
public String toString()
```

Converts this portion of the attachment to a `String`. Overrides: `toString` in class `Object`.

#### *Overrides*

`Object.toString()`



# MessageAttachment.Factory Class

*public static final class MessageAttachment.Factory*

extends Object

Static factory class for creating new instances of MessageAttachment

---

## Hierarchy

Object  
MessageAttachment.Factory

## Enclosing interface

MessageAttachment

---

## Constructor Summary

*MessageAttachment.Factory()*

## Method Summary

```
public static
MessageAttachment newInstance(XmlObject xo)
    Creates a MessageAttachment instance from an XmlObject containing
    XML data.

public static
MessageAttachment newInstance(RawData rd)
    Creates a MessageAttachment instance from a RawData object containing
    non-XML (raw) data for which there is no MFL file.
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### MessageAttachment.Factory

```
public MessageAttachment.Factory()
```

## Method Detail

### newInstance(XmlObject) Method

```
public static MessageAttachment newInstance(XmlObject xo)
```

Creates a MessageAttachment instance from an XmlObject containing XML data.

#### Parameters

*xo*  
XmlObject containing the data.

#### Returns

*a new MessageAttachment instance*

---

### newInstance(RawData) Method

```
public static MessageAttachment newInstance(RawData rd)
```

Creates a MessageAttachment instance from a RawData object containing non-XML (raw) data for which there is no MFL file.

#### Parameters

*rd*  
RawData containing the data.

#### Returns

*a new MessageAttachment instance*

# RawData Interface

*public interface* **RawData**

This defines a Java interface for a potentially large piece of data (incoming message stored in the repository) which can be a process variable. The main purpose for using this interface (as opposed to a Java byte array) is to avoid having to read the entire message into memory to perform useful operations (XT transformations, forwarding, etc) on it.

## Nested Class Summary

```

    public static final class RawData.Factory
        Static factory class for creating new instances of RawData

```

## Method Summary

```

    public byte[] byteValue()
        Returns the data stored as a byte array

    public InputStream newInputStream()
        Returns more effecient access to the data stored in this
        object.

    public String stringValue()
        Serializes the Binary object into a String.

    public String stringValue(String enc)
        Serializes the data with a specified Java encoding.

```

## Method Detail

### byteValue() Method

```

public byte[] byteValue()
    throws RawDataRuntimeException

```

Returns the data stored as a byte array

### Returns

*the bytes contained in this data object.*

## Exceptions

*RawDataRuntimeException*

---

### newInputStream() Method

```
public InputStream newInputStream()  
    throws IOException
```

Returns more efficient access to the data stored in this object.

#### Returns

*an InputStream which contains the data.*

#### Exceptions

*IOException*

---

### stringValue() Method

```
public String stringValue()  
    throws RawDataRuntimeException
```

Serializes the Binary object into a String. This uses the default encoding to convert the bytes to a String. May return a garbage string if the data is not encoded correctly.

#### Returns

*the String representation of the Binary document.*

#### Exceptions

*RawDataRuntimeException*

---

### stringValue(String) Method

```
public String stringValue(String enc)  
    throws UnsupportedOperationException, RawDataRuntimeException
```

Serializes the data with a specified Java encoding.

#### Returns

*the String representation of the Binary document.*

## Exceptions

*UnsupportedEncodingException*

*RawDataRuntimeException*

# RawData.Factory Class

*public static final class **RawData.Factory***

extends Object

Static factory class for creating new instances of RawData

---

## **Hierarchy**

Object  
RawData.Factory

## **Enclosing interface**

RawData

---

## Constructor Summary

**RawData.Factory()**

## Method Summary

```
public
static newInstance(InputStream stream)
RawData      creates a rawdata instance from inputstream
public
static newInstance(byte[] bytes)
RawData      creates a rawdata instance from byte[]
public
static newInstance(String str)
RawData      creates a rawdata instance from string
public
static newInstance(String str, String enc)
RawData      creates a rawdata instance from string with encoding
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait



## Constructor Detail

### RawData.Factory

```
public RawData.Factory()
```

## Method Detail

### newInstance(InputStream) Method

```
public static RawData newInstance(InputStream stream)
    throws IOException
```

creates a rawdata instance from inputstream

#### Parameters

*stream*  
InputStream containing the data.

#### Returns

*a new RawData instance*

#### Exceptions

*IOException*  
thrown if a reading error occurs.

---

### newInstance(byte[]) Method

```
public static RawData newInstance(byte[] bytes)
    throws RawDataRuntimeException
```

creates a rawdata instance from byte[]

#### Returns

*a new RawData instance*

#### Exceptions

*RawDataRuntimeException*

---

## newInstance(String) Method

```
public static RawData newInstance(String str)
    throws RawDataRuntimeException
```

creates a rawdata instance from string

### Parameters

*str*

The string representation of the data.

### Returns

*a new RawData instance*

### Exceptions

*RawDataRuntimeException*

---

## newInstance(String, String) Method

```
public static RawData newInstance(String str,
                                   String enc)
    throws UnsupportedOperationException, RawDataRuntimeException
```

creates a rawdata instance from string with encoding

### Parameters

*str*

The string representation of the data.

*enc*

The encoding to apply on the string representation of the data.

### Returns

*a new RawData instance*

### Exceptions

*UnsupportedEncodingException*  
*RawDataRuntimeException*

# RawDataRuntimeException Class

*public class RawDataRuntimeException*

extends RuntimeException

Used to mark runtime exceptions for RawData.

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      RuntimeException  
       RawDataRuntimeException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

***RawDataRuntimeException()***

empty exception

***RawDataRuntimeException(String message, Throwable cause)***

exception with specified detail message and cause.

***RawDataRuntimeException(String message)***

exception with specified detail message.

***RawDataRuntimeException(Throwable cause)***

exception with specified cause.

## **Method Summary**

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `RawDataRuntimeException`

```
public RawDataRuntimeException()
```

empty exception

---

### `RawDataRuntimeException`

```
public RawDataRuntimeException(String message,  
                               Throwable cause)
```

exception with specified detail message and cause.

---

### `RawDataRuntimeException`

```
public RawDataRuntimeException(String message)
```

exception with specified detail message.

---

### `RawDataRuntimeException`

```
public RawDataRuntimeException(Throwable cause)
```

exception with specified cause.

# com.bea.jpd Package

## Interface Summary

JpdContext	Represents the execution context of a business process (a JPD).
JpdContext.Callback	
JpdContext.ExceptionInfo	
ProcessDefinition	marker interface for process definitions

# JpdContext Interface

*public interface JpdContext*

extends JwsContext

Represents the execution context of a business process (a JPD). Methods in this interface can be used to access container services for JPD (Process Definition for Java) files. A JpdContext instance named `context` is included by default in each business process you create in WebLogic Workshop.

---

## All Superinterfaces

Context, JwsContext, Serializable

---

## Nested Class Summary

```

public
static JpdContext.Callback
interface
    public
    static JpdContext.ExceptionInfo
interface

```

### Nested classes from interface weblogic.jws.control.Context

Context.Callback

### Nested classes from interface com.bea.control.JwsContext

JwsContext.Callback

## Method Summary

```

public
JpdContext.ExceptionInfo getExceptionInfo()
    public String
        getInstanceId()
            Retrieves the instance id of the associated business process.
    public String
        getProcessLabel()
            Retrieves the current process label associated with this business
            process.
    public boolean

```

***isTransactionRetried()***

Indicates whether the current transaction is executing a retry.

public void

***setProcessLabel(String label)***

Associates the specified label with this instance of the business process.

public void

***trackData(XmlObject value)***

Logs the specified XML value to the log tables such that it can be correlated to a node in your business process.

public void

***trackData(RawData value)***

Logs the specified RawData value to the log tables such that it can be correlated to a node in your business process.

public void

***trackData(String value)***

Logs the specified String value to the log tables such that it can be correlated to a node in your business process.

**Methods from interface `weblogic.jws.control.Context`**

`finishConversation`, `getCallerPrincipal`, `getCurrentAge`,  
`getCurrentIdleTime`, `getLogger`, `getMaxAge`, `getMaxIdleTime`, `getService`,  
`isCallerInRole`, `isFinished`, `resetIdleTime`, `setMaxAge`, `setMaxAge`,  
`setMaxIdleTime`, `setMaxIdleTime`

**Methods from interface `com.bea.control.JwsContext`**

`getCallbackLocation`, `getCallbackPassword`, `getCallbackUsername`,  
`getInputHeaders`, `getProtocol`, `getUnderstoodInputHeaders`,  
`setCallbackLocation`, `setCallbackLocation`, `setCallbackPassword`,  
`setCallbackUsername`, `setOutputHeaders`, `setUnderstoodInputHeaders`

## Method Detail

### **getExceptionInfo() Method**

```
public JpdContext.ExceptionInfo getExceptionInfo()
```

#### **Returns**

*A handle to the exception (handler/block/instance) information.*

---

### **getInstanceId() Method**

```
public String getInstanceId()
```

Retrieves the instance id of the associated business process.

## **getProcessLabel() Method**

```
public String getProcessLabel()
```

Retrieves the current process label associated with this business process.

---

## **isTransactionRetried() Method**

```
public boolean isTransactionRetried()
```

Indicates whether the current transaction is executing a retry.

---

## **setProcessLabel(String) Method**

```
public void setProcessLabel(String label)
```

Associates the specified label with this instance of the business process. The label can be used as a business handle for an instance of a business process. For example, you can use an order number, customer number, DUNS number, or some other value of use in auditing.

---

## **trackData(XmlObject) Method**

```
public void trackData(XmlObject value)
```

Logs the specified XML value to the log tables such that it can be correlated to a node in your business process.

---

## **trackData(RawData) Method**

```
public void trackData(RawData value)
```

Logs the specified RawData value to the log tables such that it can be correlated to a node in your business process.

---

## **trackData(String) Method**

```
public void trackData(String value)
```

Logs the specified String value to the log tables such that it can be correlated to a node in your business process.



## JpdContext.Callback Interface

*public static interface JpdContext.Callback*

extends JwsContext.Callback

---

### *All Superinterfaces*

Context.Callback, JwsContext.Callback

### *Enclosing interface*

JpdContext

---

## Method Summary

### Methods from interface `weblogic.jws.control.Context.Callback`

`onAgeTimeout`, `onAsyncFailure`, `onCreate`, `onException`, `onFinish`,  
`onIdleTimeout`

# JpdContext.ExceptionInfo Interface

*public static interface JpdContext.ExceptionInfo*

*Enclosing interface*

JpdContext

---

## Method Summary

```
public
Exception getException()
    This method is only valid in an exception handler.

public
String getExceptionNodeName()
    This method is only valid in an exception handler.

public
int getRetriesRemaining()
    This method is only valid within an exception handler, or in a block of code that
    contains an exception handler.

public
int getRetryCount()
    This method is only valid within an exception handler, or in a block of code that
    contains an exception handler.
```

## Method Detail

### getException() Method

```
public Exception getException()
    throws IllegalStateException
```

This method is only valid in an exception handler.

### Returns

*The current process exception.*

### Exceptions

*IllegalStateException*  
if not currently in an exception handler.

---

## getExceptionNodeName() Method

```
public String getExceptionNodeName()  
    throws IllegalStateException
```

This method is only valid in an exception handler.

### Returns

*the name of the node that threw the exception, or returns null if the node is undefined.*

### Exceptions

*IllegalStateException*  
if not currently in an exception handler.

---

## getRetriesRemaining() Method

```
public int getRetriesRemaining()  
    throws IllegalStateException
```

This method is only valid within an exception handler, or in a block of code that contains an exception handler.

### Returns

*The number of retries remaining for the current exception block.*

### Exceptions

*IllegalStateException*  
if not currently in an exception block or handler.

---

## getRetryCount() Method

```
public int getRetryCount()  
    throws IllegalStateException
```

This method is only valid within an exception handler, or in a block of code that contains an exception handler.

### Returns

*The number of times the current exception block has been retried.*

### Exceptions

*IllegalStateException*  
if not currently in an exception block or handler.



com.bea.jpdp

# ProcessDefinition Interface

*public interface **ProcessDefinition***

extends Serializable

marker interface for process definitions

---

## *All Superinterfaces*

Serializable

---

# com.bea.transform Package

## Interface Summary

TransformFaultCodes

TransformSource      marker interface for a transform source file

## Exception Summary

TransformException    Exception for Transformation runtime exceptions.

# TransformException Class

*public class TransformException*

extends RuntimeException

Exception for Transformation runtime exceptions.

---

## *Hierarchy*

Object  
  Throwable  
    Exception  
      RuntimeException  
        TransformException

## *All Implemented Interfaces*

Serializable

---

## Field Summary

protected int

**\_code**  
        the code

protected String

**\_message**  
        a possible user  
        message

protected Object

**\_params**  
        the parameters

## Constructor Summary

*TransformException(XQRLEException e)*

XQRLEException wrapper

*TransformException(int code, Object param1, Object param2, Throwable cause)*

***TransformException(int code, Object param1, Object param2)***

***TransformException(int code, Object param, Throwable cause)***

***TransformException(int code, Object param)***

***TransformException(int code, Object params, Throwable cause)***

***TransformException(int code, Object params)***

***TransformException(int code, Throwable cause)***

***TransformException(int code)***

constructors for system defined exceptions

***TransformException(String message, Object param1, Object param2, Throwable cause)***

***TransformException(String message, Object param1, Object param2)***

***TransformException(String message, Object param, Throwable cause)***

***TransformException(String message, Object param)***

***TransformException(String message, Object params, Throwable cause)***

***TransformException(String message, Object params)***

***TransformException(String message, Throwable cause)***

***TransformException(String message)***



constructors for user defined exceptions

## Method Summary

```
public
    int getFaultCode()
        Returns the fault code of the transform exception For details see
        TransformFaultCodes.java

public
    String getMessage()
        exception message
```

### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace,
initCause, printStackTrace, printStackTrace, printStackTrace,
setStackTrace, toString
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Field Detail

### **`_code`**

```
protected int _code
```

the code

---

### **`_message`**

```
protected String _message
```

a possible user message

---

### **`_params`**

```
protected Object _params
```

the parameters

## Constructor Detail

### TransformException

```
public TransformException(XQRLException e)
```

XQRLException wrapper

---

### TransformException

```
public TransformException(int code,  
                           Object param1,  
                           Object param2,  
                           Throwable cause)
```

---

### TransformException

```
public TransformException(int code,  
                           Object param1,  
                           Object param2)
```

---

### TransformException

```
public TransformException(int code,  
                           Object param,  
                           Throwable cause)
```

---

### TransformException

```
public TransformException(int code,  
                           Object param)
```

---

### TransformException

```
public TransformException(int code,  
                           Object[] params,  
                           Throwable cause)
```

---

### TransformException

```
public TransformException(int code,  
                           Object[] params)
```

---

## TransformException

```
public TransformException(int code,  
                          Throwable cause)
```

---

## TransformException

```
public TransformException(int code)
```

constructors for system defined exceptions

---

## TransformException

```
public TransformException(String message,  
                          Object param1,  
                          Object param2,  
                          Throwable cause)
```

---

## TransformException

```
public TransformException(String message,  
                          Object param1,  
                          Object param2)
```

---

## TransformException

```
public TransformException(String message,  
                          Object param,  
                          Throwable cause)
```

---

## TransformException

```
public TransformException(String message,  
                          Object param)
```

---

## TransformException

```
public TransformException(String message,  
                          Object[] params,  
                          Throwable cause)
```

---

## TransformException

```
public TransformException(String message,  
                          Object[] params)
```

## TransformException

```
public TransformException(String message,  
                           Throwable cause)
```

---

## TransformException

```
public TransformException(String message)
```

constructors for user defined exceptions

## Method Detail

### getFaultCode() Method

```
public int getFaultCode()
```

Returns the fault code of the transform exception For details see TransformFaultCodes.java

---

### getMessage() Method

```
public String getMessage()
```

exception message

#### *Overrides*

Throwable.getMessage()

# TransformFaultCodes Interface

*public interface TransformFaultCodes*

## Field Summary

```

    public
    static INCORRECT_TRANSFORM_CONFIG
    final int          Load time errors for dev mode

    public
    static INTERNAL_ERROR
    final int          int

    public
    static INVALID_XML
    final int          int

    public
    static NO_CODE_MSG
    final int          generic exceptions

    public
    static RT_BEV_FAILED
    final int          int

    public
    static RT_CANNOT_COMPARE
    final int          int

    public
    static RT_CLOSED_MAT_QUERY
    final int          int

    public
    static RT_DIV_ZERO
    final int          int

    public
    static RT_DOUBLE_CLOSE
    final int          int

    public
    static RT_DOUBLE_OPEN
    final int          int

    public
    static RT_DUP_ATTR_DEF
    final int          int

    public
    static RT_EMPTY_ATTR
    final int          int

    public

```

```

    static RT_EXPECTED_TOKEN
final int      int

    public
    static RT_EXT_METHOD
final int      int

    public
    static RT_GETTYPE_ON_OPEN
final int      int

    public
    static RT_ILLEGAL_CAST
final int      int

    public
    static RT_ILLEGAL_DATA_ARG
final int      int

    public
    static RT_ILLEGAL_INDEX
final int      int

    public
    static RT_ILLEGAL_PATH
final int      int

    public
    static RT_ILLEGAL_SEQ
final int      int

    public
    static RT_ILLEGAL_URI
final int      int

    public
    static RT_INVALID_ARGUMENT
final int      int

    public
    static RT_INVALID_DOC_URI
final int      int

    public
    static RT_JAVA_CONVERSION
final int      int

    public
    static RT_LOC_ITERATOR
final int      int

    public
    static RT_MAX_REC_DEPTH
final int      int

    public
    static RT_MIXED_SEQ
final int      int

    public
    static RT_NO_ARITH
final int      int

```

```

    public
    static RT_NO_OPEN
final int      int

    public
    static RT_NORMAL_FORM
final int      int

    public
    static RT_NOT_SUBTYPE
final int      int

    public
    static RT_NULL_CONVERSION
final int      int

    public
    static RT_PARENT_ROOT
final int      int

    public
    static RT_QUANT_MISMATCH
final int      int

    public
    static RT_REGEXP
final int      int

    public
    static RT_REGEXP_FLAGS
final int      int

    public
    static RT_TYPED_VALUE
final int      int

    public
    static RT_UNEXPECTED_END
final int      int

    public
    static RT_UNKNOWN_TYPE
final int      int

    public
    static RT_VALUE_INSTEAD_NODE
final int      int

    public
    static RT_XQUERY_ERROR
final int      int

    public
    static SYS_ASSERT
final int      int

    public
    static SYS_BAD_QNAME_INITIALIZER
final int      int

    public
    static SYS_BAD_USAGE
final int      int

```

```

    public
    static SYS_CAUGHT_EX
final int      int

    public
    static SYS_CLONE_ERROR
final int      int

    public
    static SYS_CLOSE_NOOPEN
final int      int

    public
    static SYS_DOCUMENT_URI
final int      int

    public
    static SYS_DOUBLE_OPEN
final int      int

    public
    static SYS_EXPECTED_ELEMENT
final int      int

    public
    static SYS_EXPECTED_ITEM
final int      int

    public
    static SYS_EXPECTED_QNAME
final int      int

    public
    static SYS_EXPECTED_STRING
final int      int

    public
    static SYS_EXPECTED_TYPE
final int      int

    public
    static SYS_EXPECTED_VALUE
final int      int

    public
    static SYS_FETCH_AFTEREND
final int      int

    public
    static SYS_FETCH_AFTERERROR
final int      int

    public
    static SYS_FETCH_NOOPEN
final int      int

    public
    static SYS_ID_EXPECTED
final int      int

    public
    static SYS_ILLEGAL_COMPARISON
final int      int

```



```

    public
    static SYS_ILLEGAL_CONTENT
final int        int

    public
    static SYS_ILLEGAL_EMPTY_SEQ
final int        int

    public
    static SYS_ILLEGAL_INDEX_USE
final int        int

    public
    static SYS_ILLEGAL_NESTING
final int        int

    public
    static SYS_ILLEGAL_SEQ
final int        int

    public
    static SYS_ILLEGAL_STATE
final int        int

    public
    static SYS_ILLEGAL_TEXT_NESTING
final int        int

    public
    static SYS_ILLEGAL_TEXT_SEQ
final int        int

    public
    static SYS_ILLEGAL_USE_ID
final int        int

    public
    static SYS_INTERNAL
final int        int

    public
    static SYS_ITEM_NOT_FOUND
final int        int

    public
    static SYS_LONG_OVERFLOW
final int        int

    public
    static SYS_MISSING_END
final int        int

    public
    static SYS_MISSING_FCT
final int        int

    public
    static SYS_MISSING_INDEX
final int        int

    public
    static SYS_MISSING_TYPE
final int        int

```

```

    public
    static SYS_NO_ARGS
final int      int

    public
    static SYS_NO_NODE_TOKEN
final int      int

    public
    static SYS_NONBEGIN_HAS_ID
final int      int

    public
    static SYS_NOT_SERIALIZED
final int      int

    public
    static SYS_NOTIMPLEMENTED
final int      int

    public
    static SYS_OP_REDEFINED
final int      int

    public
    static SYS_OPR_CREATE_FAILED
final int      int

    public
    static SYS_REGISTER_MISMATCH
final int      int

    public
    static SYS_RESET_CONTEXT
final int      int

    public
    static SYS_SKIP_EMPTY
final int      int

    public
    static SYS_TOKEN_STREAM
final int      int

    public
    static SYS_TYPE_CAST_BOOLEAN
final int      int

    public
    static SYS_TYPE_GENERAL
final int      int

    public
    static SYS_UNEXPECTED_END
final int      int

    public
    static SYS_UNEXPECTED_QNAME
final int      int

    public
    static SYS_UNEXPECTED_TOKEN
final int      int

```

```

    public
    static SYS_UNEXPECTED_TYPE
final int      int

    public
    static SYS_VALUE_INSTEAD_NODE
final int      int

    public
    static SYS_WRONG_ARITY
final int      int

    public
    static TOPLEVELQNAME_MISMATCH
final int      int

    public
    static UNEXPECTED_ERROR
final int      int

    public
    static XJM_CAST_EX
final int      int

    public
    static XJM_COLLECTION_ERROR
final int      int

    public
    static XJM_DUPLICATE_FIELD
final int      int

    public
    static XJM_FIELD_ACCESS
final int      int

    public
    static XJM_OBJECT_CREATION
final int      XML Java mapping exceptions

    public
    static XJM_UNKNOWN_FIELD
final int      int

    public
    static XJM_UNSUPPORTED_NODE
final int      int

    public
    static XQRL_UNEXPECTED_ERROR
final int      XQRL exceptions

    public
    static XQUERY_EXTERNAL_FUNCTION_EXCEPTION
final int      int

    public
    static XQUERY_EXTERNAL_FUNCTION_NOT_FOUND
final int      XQuery external function error

    public
    static XQUERY_EXTERNAL_FUNCTION_PARAM_MISMATCH
final int      int

```

```

    public
    static XQUERY_EXTERNAL_FUNCTION_PARAM_UNMARSHAL
final int      int

    public
    static XQUERY_EXTERNAL_FUNCTION_RESULT_MARSHAL
final int      int

    public
    static XSLT_RUNTIME_EX
final int      XSLT exceptions

    public
    static XT_RUNTIME_EX
final int      XT exceptions

```

## Field Detail

### INCORRECT\_TRANSFORM\_CONFIG

```
public static final int INCORRECT_TRANSFORM_CONFIG
```

Load time errors for dev mode

---

### INTERNAL\_ERROR

```
public static final int INTERNAL_ERROR
```

---

### INVALID\_XML

```
public static final int INVALID_XML
```

---

### NO\_CODE\_MSG

```
public static final int NO_CODE_MSG
```

generic exceptions

---

### RT\_BEV\_FAILED

```
public static final int RT_BEV_FAILED
```

---

### RT\_CANNOT\_COMPARE

```
public static final int RT_CANNOT_COMPARE
```

---

**RT\_CLOSED\_MAT\_QUERY**

```
public static final int RT_CLOSED_MAT_QUERY
```

---

**RT\_DIV\_ZERO**

```
public static final int RT_DIV_ZERO
```

---

**RT\_DOUBLE\_CLOSE**

```
public static final int RT_DOUBLE_CLOSE
```

---

**RT\_DOUBLE\_OPEN**

```
public static final int RT_DOUBLE_OPEN
```

---

**RT\_DUP\_ATTR\_DEF**

```
public static final int RT_DUP_ATTR_DEF
```

---

**RT\_EMPTY\_ATTR**

```
public static final int RT_EMPTY_ATTR
```

---

**RT\_EXPECTED\_TOKEN**

```
public static final int RT_EXPECTED_TOKEN
```

---

**RT\_EXT\_METHOD**

```
public static final int RT_EXT_METHOD
```

---

**RT\_GETTYPE\_ON\_OPEN**

```
public static final int RT_GETTYPE_ON_OPEN
```

---

**RT\_ILLEGAL\_CAST**

```
public static final int RT_ILLEGAL_CAST
```

---

**RT\_ILLEGAL\_DATA\_ARG**

```
public static final int RT_ILLEGAL_DATA_ARG
```

---

**RT\_ILLEGAL\_INDEX**

```
public static final int RT_ILLEGAL_INDEX
```

---

**RT\_ILLEGAL\_PATH**

```
public static final int RT_ILLEGAL_PATH
```

---

**RT\_ILLEGAL\_SEQ**

```
public static final int RT_ILLEGAL_SEQ
```

---

**RT\_ILLEGAL\_URI**

```
public static final int RT_ILLEGAL_URI
```

---

**RT\_INVALID\_ARGUMENT**

```
public static final int RT_INVALID_ARGUMENT
```

---

**RT\_INVALID\_DOC\_URI**

```
public static final int RT_INVALID_DOC_URI
```

---

**RT\_JAVA\_CONVERSION**

```
public static final int RT_JAVA_CONVERSION
```

---

**RT\_LOC\_ITERATOR**

```
public static final int RT_LOC_ITERATOR
```

---

**RT\_MAX\_REC\_DEPTH**

```
public static final int RT_MAX_REC_DEPTH
```

---

**RT\_MIXED\_SEQ**

```
public static final int RT_MIXED_SEQ
```

---

**RT\_NO\_ARITH**

```
public static final int RT_NO_ARITH
```

---

**RT\_NO\_OPEN**

```
public static final int RT_NO_OPEN
```

---

**RT\_NORMAL\_FORM**

```
public static final int RT_NORMAL_FORM
```

---

**RT\_NOT\_SUBTYPE**

```
public static final int RT_NOT_SUBTYPE
```

---

**RT\_NULL\_CONVERSION**

```
public static final int RT_NULL_CONVERSION
```

---

**RT\_PARENT\_ROOT**

```
public static final int RT_PARENT_ROOT
```

---

**RT\_QUANT\_MISMATCH**

```
public static final int RT_QUANT_MISMATCH
```

---

**RT\_REGEXP**

```
public static final int RT_REGEXP
```

---

**RT\_REGEXP\_FLAGS**

```
public static final int RT_REGEXP_FLAGS
```

---

**RT\_TYPED\_VALUE**

```
public static final int RT_TYPED_VALUE
```

---

**RT\_UNEXPECTED\_END**

```
public static final int RT_UNEXPECTED_END
```

---

**RT\_UNKNOWN\_TYPE**

```
public static final int RT_UNKNOWN_TYPE
```

---

**RT\_VALUE\_INSTEAD\_NODE**

```
public static final int RT_VALUE_INSTEAD_NODE
```

---

**RT\_XQUERY\_ERROR**

```
public static final int RT_XQUERY_ERROR
```

---

**SYS\_ASSERT**

```
public static final int SYS_ASSERT
```

---

**SYS\_BAD\_QNAME\_INITIALIZER**

```
public static final int SYS_BAD_QNAME_INITIALIZER
```

---

**SYS\_BAD\_USAGE**

```
public static final int SYS_BAD_USAGE
```

---

**SYS\_CAUGHT\_EX**

```
public static final int SYS_CAUGHT_EX
```

---

**SYS\_CLONE\_ERROR**

```
public static final int SYS_CLONE_ERROR
```

---



**SYS\_CLOSE\_NOOPEN**

```
public static final int SYS_CLOSE_NOOPEN
```

---

**SYS\_DOCUMENT\_URI**

```
public static final int SYS_DOCUMENT_URI
```

---

**SYS\_DOUBLE\_OPEN**

```
public static final int SYS_DOUBLE_OPEN
```

---

**SYS\_EXPECTED\_ELEMENT**

```
public static final int SYS_EXPECTED_ELEMENT
```

---

**SYS\_EXPECTED\_ITEM**

```
public static final int SYS_EXPECTED_ITEM
```

---

**SYS\_EXPECTED\_QNAME**

```
public static final int SYS_EXPECTED_QNAME
```

---

**SYS\_EXPECTED\_STRING**

```
public static final int SYS_EXPECTED_STRING
```

---

**SYS\_EXPECTED\_TYPE**

```
public static final int SYS_EXPECTED_TYPE
```

---

**SYS\_EXPECTED\_VALUE**

```
public static final int SYS_EXPECTED_VALUE
```

---

**SYS\_FETCH\_AFTEREND**

```
public static final int SYS_FETCH_AFTEREND
```

---

**SYS\_FETCH\_AFTERERROR**

```
public static final int SYS_FETCH_AFTERERROR
```

---

**SYS\_FETCH\_NOOPEN**

```
public static final int SYS_FETCH_NOOPEN
```

---

**SYS\_ID\_EXPECTED**

```
public static final int SYS_ID_EXPECTED
```

---

**SYS\_ILLEGAL\_COMPARISON**

```
public static final int SYS_ILLEGAL_COMPARISON
```

---

**SYS\_ILLEGAL\_CONTENT**

```
public static final int SYS_ILLEGAL_CONTENT
```

---

**SYS\_ILLEGAL\_EMPTY\_SEQ**

```
public static final int SYS_ILLEGAL_EMPTY_SEQ
```

---

**SYS\_ILLEGAL\_INDEX\_USE**

```
public static final int SYS_ILLEGAL_INDEX_USE
```

---

**SYS\_ILLEGAL\_NESTING**

```
public static final int SYS_ILLEGAL_NESTING
```

---

**SYS\_ILLEGAL\_SEQ**

```
public static final int SYS_ILLEGAL_SEQ
```

---

**SYS\_ILLEGAL\_STATE**

```
public static final int SYS_ILLEGAL_STATE
```

---

**SYS\_ILLEGAL\_TEXT\_NESTING**

```
public static final int SYS_ILLEGAL_TEXT_NESTING
```

---

**SYS\_ILLEGAL\_TEXT\_SEQ**

```
public static final int SYS_ILLEGAL_TEXT_SEQ
```

---

**SYS\_ILLEGAL\_USE\_ID**

```
public static final int SYS_ILLEGAL_USE_ID
```

---

**SYS\_INTERNAL**

```
public static final int SYS_INTERNAL
```

---

**SYS\_ITEM\_NOT\_FOUND**

```
public static final int SYS_ITEM_NOT_FOUND
```

---

**SYS\_LONG\_OVERFLOW**

```
public static final int SYS_LONG_OVERFLOW
```

---

**SYS\_MISSING\_END**

```
public static final int SYS_MISSING_END
```

---

**SYS\_MISSING\_FCT**

```
public static final int SYS_MISSING_FCT
```

---

**SYS\_MISSING\_INDEX**

```
public static final int SYS_MISSING_INDEX
```

---

**SYS\_MISSING\_TYPE**

```
public static final int SYS_MISSING_TYPE
```

---

**SYS\_NO\_ARGS**

```
public static final int SYS_NO_ARGS
```

---

**SYS\_NO\_NODE\_TOKEN**

```
public static final int SYS_NO_NODE_TOKEN
```

---

**SYS\_NONBEGIN\_HAS\_ID**

```
public static final int SYS_NONBEGIN_HAS_ID
```

---

**SYS\_NOT\_SERIALIZED**

```
public static final int SYS_NOT_SERIALIZED
```

---

**SYS\_NOTIMPLEMENTED**

```
public static final int SYS_NOTIMPLEMENTED
```

---

**SYS\_OP\_REDEFINED**

```
public static final int SYS_OP_REDEFINED
```

---

**SYS\_OPR\_CREATE\_FAILED**

```
public static final int SYS_OPR_CREATE_FAILED
```

---

**SYS\_REGISTER\_MISMATCH**

```
public static final int SYS_REGISTER_MISMATCH
```

---

**SYS\_RESET\_CONTEXT**

```
public static final int SYS_RESET_CONTEXT
```

---

**SYS\_SKIP\_EMPTY**

```
public static final int SYS_SKIP_EMPTY
```

---

**SYS\_TOKEN\_STREAM**

```
public static final int SYS_TOKEN_STREAM
```

---

**SYS\_TYPE\_CAST\_BOOLEAN**

```
public static final int SYS_TYPE_CAST_BOOLEAN
```

---

**SYS\_TYPE\_GENERAL**

```
public static final int SYS_TYPE_GENERAL
```

---

**SYS\_UNEXPECTED\_END**

```
public static final int SYS_UNEXPECTED_END
```

---

**SYS\_UNEXPECTED\_QNAME**

```
public static final int SYS_UNEXPECTED_QNAME
```

---

**SYS\_UNEXPECTED\_TOKEN**

```
public static final int SYS_UNEXPECTED_TOKEN
```

---

**SYS\_UNEXPECTED\_TYPE**

```
public static final int SYS_UNEXPECTED_TYPE
```

---

**SYS\_VALUE\_INSTEAD\_NODE**

```
public static final int SYS_VALUE_INSTEAD_NODE
```

---

**SYS\_WRONG\_ARITY**

```
public static final int SYS_WRONG_ARITY
```

---

**TOPLEVELQNAME\_MISMATCH**

```
public static final int TOPLEVELQNAME_MISMATCH
```

---

**UNEXPECTED\_ERROR**

```
public static final int UNEXPECTED_ERROR
```

---

**XJM\_CAST\_EX**

```
public static final int XJM_CAST_EX
```

---

**XJM\_COLLECTION\_ERROR**

```
public static final int XJM_COLLECTION_ERROR
```

---

**XJM\_DUPLICATE\_FIELD**

```
public static final int XJM_DUPLICATE_FIELD
```

---

**XJM\_FIELD\_ACCESS**

```
public static final int XJM_FIELD_ACCESS
```

---

**XJM\_OBJECT\_CREATION**

```
public static final int XJM_OBJECT_CREATION
```

XML Java mapping exceptions

---

**XJM\_UNKNOWN\_FIELD**

```
public static final int XJM_UNKNOWN_FIELD
```

---

**XJM\_UNSUPPORTED\_NODE**

```
public static final int XJM_UNSUPPORTED_NODE
```

---

**XQRL\_UNEXPECTED\_ERROR**

```
public static final int XQRL_UNEXPECTED_ERROR
```

XQRL exceptions

---

## **XQUERY\_EXTERNAL\_FUNCTION\_EXCEPTION**

```
public static final int XQUERY_EXTERNAL_FUNCTION_EXCEPTION
```

---

## **XQUERY\_EXTERNAL\_FUNCTION\_NOT\_FOUND**

```
public static final int XQUERY_EXTERNAL_FUNCTION_NOT_FOUND
```

XQuery external function error

---

## **XQUERY\_EXTERNAL\_FUNCTION\_PARAM\_MISMATCH**

```
public static final int XQUERY_EXTERNAL_FUNCTION_PARAM_MISMATCH
```

---

## **XQUERY\_EXTERNAL\_FUNCTION\_PARAM\_UNMARSHAL**

```
public static final int XQUERY_EXTERNAL_FUNCTION_PARAM_UNMARSHAL
```

---

## **XQUERY\_EXTERNAL\_FUNCTION\_RESULT\_MARSHAL**

```
public static final int XQUERY_EXTERNAL_FUNCTION_RESULT_MARSHAL
```

---

## **XSLT\_RUNTIME\_EX**

```
public static final int XSLT_RUNTIME_EX
```

XSLT exceptions

---

## **XT\_RUNTIME\_EX**

```
public static final int XT_RUNTIME_EX
```

XT exceptions

com.bea.transform

# TransformSource Interface

*public interface TransformSource*

extends Serializable

marker interface for a transform source file

---

## *All Superinterfaces*

Serializable

---



# com.bea.wli.bpm.proxy Package

## Interface Summary

JpdProxy.ContextHandler Clients pass an instance of this interface to the `create` method.

JpdProxySession Jpd proxies implement this interface.

## Class Summary

JpdProxy Factory class for WebLogic Integration Processes (jpd files).

## Exception Summary

JpdProxyException

# JpdProxy Class

*public abstract class JpdProxy*

extends Object  
implements Serializable

Factory class for WebLogic Integration Processes (jpd files). Clients call one of the create methods to get a proxy instance. The create methods take the Class of the jpd's public contract. The return value of the create method can be cast to the public contract interface.

---

## ***Hierarchy***

Object  
JpdProxy

## ***All Implemented Interfaces***

Serializable

---

## **Nested Class Summary**

public static interface **JpdProxy.ContextHandler**  
Clients pass an instance of this interface to the create method.

## **Constructor Summary**

*JpdProxy()*

## **Method Summary**

public static final Object **create**(Class publicContract, String serviceUri, JpdProxy.ContextHandler ch)  
Creates a client proxy for a WebLogic Integration Process (jpd).  
public static final Object **create**(Class publicContract, String serviceUri)  
Creates a client proxy for a WebLogic Integration Process (jpd).

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

**Constructor Detail****JpdProxy**

```
public JpdProxy()
```

**Method Detail****`create(Class, String, JpdProxy.ContextHandler)` Method**

```
public static final Object create(Class publicContract,
                                String serviceUri,
                                JpdProxy.ContextHandler ch)
    throws JpdProxyException
```

Creates a client proxy for a WebLogic Integration Process (jpd). A public contract interface that describes the methods of the jpd is passed in. The result of this call can be typecast to the public contract class. The `JpdProxy.ContextHandler` is invoked by the proxy to obtain the JNDI context used to login to the server and lookup server-side resources. The `serviceUri` uniquely identifies the jpd on the server.

**Parameters**

*publicContract*  
the jpd's public contract interface

*serviceUri*  
the uri of the jpd

*ch*  
a context handler, see `JpdProxy.ContextHandler`

**Returns**

*a proxy object that can be cast to the public contract interface*

**Exceptions**

*JpdProxyException*  
This exception wraps any checked exceptions thrown during construction of the proxy.

## create(Class, String) Method

```
public static final Object create(Class publicContract,  
                                String serviceUri)  
    throws JpdProxyException
```

Creates a client proxy for a WebLogic Integration Process (jpd). A public contract interface that describes the methods of the jpd is passed in. The result of this call can be typecast to the public contract class. The serviceUri uniquely identifies the jpd on the server. Use this method when the client is running on the same WLS domain as the target jpd.

### Parameters

*publicContract*  
the jpd's public contract interface

*serviceUri*  
the uri of the jpd

### Returns

*a proxy object that can be cast to the public contract interface*

### Exceptions

*JpdProxyException*  
This exception wraps any checked exceptions thrown during construction of the proxy.

# JpdProxy.ContextHandler Interface

*public static interface JpdProxy.ContextHandler*

extends `Serializable`

Clients pass an instance of this interface to the `create` method. The proxy implementation uses this instance at runtime to allocate a JNDI context. This context is used to login to the server and lookup server-side resources that handle incoming proxy requests.

---

## *All Superinterfaces*

`Serializable`

## *Enclosing class*

`JpdProxy`

---

## Field Summary

```
public
static serialVersionUID
final long long
```

## Method Summary

```
public
Context getContext()
```

Returns a JNDI context used to lookup the server-side bean that this proxy talks to.

## Field Detail

### **serialVersionUID**

```
public static final long serialVersionUID
```

## Method Detail

## getContext() Method

```
public Context getContext()  
    throws NamingException
```

Returns a JNDI context used to lookup the server-side bean that this proxy talks to.

### Returns

*a JNDI context, must be properly configured to connect to the server*

### Exceptions

*NamingException*  
if the context could not be created

# JpdProxyException Class

*public class JpdProxyException*

extends Exception

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      JpdProxyException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

*JpdProxyException()*

Constructs a new exception with null as its detail message.

*JpdProxyException(String message, Throwable cause)*

Constructs a new exception with the specified detail message and cause.

*JpdProxyException(String message)*

Constructs a new exception with the specified detail message.

*JpdProxyException(Throwable cause)*

Constructs a new exception with the specified cause and a detail message of  
(cause==null ?

## **Method Summary**

Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`,  
`getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,  
`printStackTrace`, `setStackTrace`, `toString`

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### `JpdProxyException`

```
public JpdProxyException()
```

Constructs a new exception with `null` as its detail message. The cause is not initialized, and may subsequently be initialized by a call to `JpdProxyException.initCause(Throwable)`.

---

### `JpdProxyException`

```
public JpdProxyException(String message,  
                          Throwable cause)
```

Constructs a new exception with the specified detail message and cause.

Note that the detail message associated with `cause` is *not* automatically incorporated in this exception's detail message.

---

### `JpdProxyException`

```
public JpdProxyException(String message)
```

Constructs a new exception with the specified detail message. The cause is not initialized, and may subsequently be initialized by a call to `JpdProxyException.initCause(Throwable)`.

---

### `JpdProxyException`

```
public JpdProxyException(Throwable cause)
```

Constructs a new exception with the specified cause and a detail message of `(cause==null ? null : cause.toString())` (which typically contains the class and detail message of `cause`). This constructor is useful for exceptions that are little more than wrappers for other throwables (for example, ).



# JpdProxySession Interface

*public interface JpdProxySession*

Jpd proxies implement this interface. Clients can type-cast proxies returned by `JpdProxy.create()` to `JpdProxySession` to set/get the conversation id used in jpd invocations.

## Method Summary

```
public
String getConversationID()
    Returns the current conversation id in use by the
    proxy.

public
    void reset()
        Resets the conversation id back to null.

public
    void setConversationID(String conversationID)
        Sets the conversation id for future invocations.
```

## Method Detail

### getConversationID() Method

```
public String getConversationID()
```

Returns the current conversation id in use by the proxy.

#### Returns

*a conversation id*

---

### reset() Method

```
public void reset()
```

Resets the conversation id back to null.

---

### setConversationID(String) Method

```
public void setConversationID(String conversationID)
```

Sets the conversation id for future invocations. If the conversation Id has not been set, or if it is set to null,

## Java Class Reference

the server will assign a unique conversation id to the jpd and will return it to the client. The proxy will then use this id in all sub-sequent calls until it is explicitly set again.

### Parameters

*conversationID*

the jpd conversation id

# com.bea.wli.calendar.api Package

## Interface Summary

BusinessCalendar

BusinessCalendarHome

## Class Summary

BusinessCalendarFactory This factory is used to create an instance of BusinessCalendar.

## Exception Summary

BusinessCalendarException

# BusinessCalendar Interface

*public interface **BusinessCalendar***

extends EJBObject

---

## *All Superinterfaces*

EJBObject, Remote

---

## Method Summary

```

public Date
    addBusinessTime(String interval, String userOrGroupId, Date
        startTime)
        Determines if userOrGroupId is a user id or group id.

public Date
    addBusinessTime(String interval, String userOrGroupId)
        Use startTime as current time.

public Date
    addBusinessTime(String interval, CalendarDocument cDoc)

public Date
    addBusinessTime(String interval, CalendarDocument cDoc, Date
        startTime)

public Date
    addBusinessTimeCalendarName(String interval, String calendarName)

public Date
    addBusinessTimeCalendarName(String interval, String calendarName,
        Date startTime)

public Date
    addBusinessTimeSystemCalendar(String interval, Date startTime)

public Date
    addExactTimeCalendarName(String interval, String calendarName,
        Date startTime)

public boolean
    calendarNameExists(String name)
        Checks if a calendar with the given name exists.

public void
    createCalendar(CalendarDocument cDoc)
        Creates a new calendar from the information passed in the
        CalendarDocument.

public void
    deleteCalendar(String calendarName)

```

```

public void deleteCalendars(String[] names)

public Date
    findFutureBusinessTime(String interval, CalendarDocument cDoc,
        Date startTime)

public Date
    findFutureBusinessTimeCalendarName(String interval, String
        calendarName, Date startTime)

public String[]
    getAllCalendarNames()

    public
CalendarDocument[] getAllCalendars()
    Gets all calendars, including rules info and mapped groups and users.

    public
CalendarDocument getCalendar(String calendarName)

    public
CalendarDocument getGroupCalendar(String groupId)

    public
CalendarDocument getGroupCalendarWalkHierarchy(String groupId)
    If a calendar is mapped to the group then that calendar is returned, otherwise
    returns the system calendar.

    public Date
        getNextFreeMinute(CalendarDocument cDoc, Date time)
        Returns the next free minute given the rules in the calendar passed from the
        time passed.

    public
CalendarDocument getSystemCalendar()

    public
CalendarDocument getUserCalendar(String userId)

    public
CalendarDocument getUserCalendarWalkHierarchy(String userId)
    If a calendar is mapped to the user then that calendar is returned, otherwise
    returns the system calendar.

    public void
        insertUserCalendarMapping(String userid, String calendarId)
        Same as updateUserCalendar.

    public boolean
        isTimeFree(CalendarDocument cDoc, Date time)

    public boolean
        isUserTimeFree(String userId, Date time)
        Looks up the user's calendar using getUserCalendarWalkHierarchy and
        checks if the time passed is free based on the rules in the calendar.

    public void
        setGroupCalendar(String groupId, String calendarName)

    public void
        setUserCalendar(String userId, String calendarName)
        Set the user's calendar.

    public Date

```

***subtractBusinessTime***(*String interval, String userOrGroupId, Date startTime*)

Go back in time by the given interval.

public Date

***subtractBusinessTime***(*String interval, String userOrGroupId*)

public Date

***subtractBusinessTime***(*String interval, CalendarDocument cDoc*)

public Date

***subtractBusinessTime***(*String interval, CalendarDocument cDoc, Date startTime*)

public Date

***subtractBusinessTimeCalendarName***(*String interval, String calendarName*)

public Date

***subtractBusinessTimeCalendarName***(*String interval, String calendarName, Date startTime*)

public Date

***subtractBusinessTimeSystemCalendar***(*String interval, Date startTime*)

public boolean

***systemCalendarExists***()

public void

***unsetGroupCalendar***(*String groupId*)

public void

***unsetUserCalendar***(*String userId*)

Unset the user's calendar.

public void

***updateCalendar***(*String oldCalendarName, CalendarDocument cDoc*)

Update the calendar – oldCalendarName, using the passed CalendarDocument.

public void

***updateSystemCalendar***(*CalendarDocument cDoc*)

Update the system calendar using the passed CalendarDocument.

## Methods from interface javax.ejb.EJBObject

getEJBHome, getHandle, getPrimaryKey, isIdentical, remove

## Method Detail

### addBusinessTime(String, String, Date) Method

```
public Date addBusinessTime(String interval,
                           String userOrGroupId,
                           Date startTime)
    throws BusinessCalendarException, RemoteException
```

Determines if `userOrGroupId` is a user id or group id. Looks up the calendar using the appropriate "walk hierarchy" method. Adds the business time interval specified by the `interval` parameter to the given `startTime` based on the rules of the calendar looked up. Note that the interval time periods are "Business times". Business time for a unit of time (days, hours, minutes) is defined as a period of the equivalent "real" time minutes which contains at least one free minute. E.g. 1 business day is a period of 24\*60 minutes which has at least one minute free.

### Parameters

*interval*

– a string in the format "d days, h hours, m minutes". e.g. "5 days", "5 days 2 minutes" Note that only days, hours, and minutes are supported.

*userOrGroupId*

*startTime*

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## addBusinessTime(String, String) Method

```
public Date addBusinessTime(String interval,
                           String userOrGroupId)
    throws BusinessCalendarException, RemoteException
```

Use `startTime` as current time.

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## addBusinessTime(String, CalendarDocument) Method

```
public Date addBusinessTime(String interval,
                           CalendarDocument cDoc)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## addBusinessTime(String, CalendarDocument, Date) Method

```
public Date addBusinessTime(String interval,
                           CalendarDocument cDoc,
                           Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## addBusinessTimeCalendarName(String, String) Method

```
public Date addBusinessTimeCalendarName(String interval,
                                       String calendarName)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## addBusinessTimeCalendarName(String, String, Date) Method

```
public Date addBusinessTimeCalendarName(String interval,
                                       String calendarName,
                                       Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## addBusinessTimeSystemCalendar(String, Date) Method

```
public Date addBusinessTimeSystemCalendar(String interval,
                                       Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---



## addExactTimeCalendarName(String, String, Date) Method

```
public Date addExactTimeCalendarName(String interval,
                                     String calendarName,
                                     Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## calendarNameExists(String) Method

```
public boolean calendarNameExists(String name)
    throws BusinessCalendarException, RemoteException
```

Checks if a calendar with the given name exists.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## createCalendar(CalendarDocument) Method

```
public void createCalendar(CalendarDocument cDoc)
    throws BusinessCalendarException, RemoteException
```

Creates a new calendar from the information passed in the CalendarDocument.

### Exceptions

*BusinessCalendarException*  
if a calendar with the given name exists or of the document is not valid.  
*RemoteException*

---

## deleteCalendar(String) Method

```
public void deleteCalendar(String calendarName)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## deleteCalendars(String[]) Method

```
public void deleteCalendars(String[] names)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## findFutureBusinessTime(String, CalendarDocument, Date) Method

```
public Date findFutureBusinessTime(String interval,
    CalendarDocument cDoc,
    Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## findFutureBusinessTimeCalendarName(String, String, Date) Method

```
public Date findFutureBusinessTimeCalendarName(String interval,
    String calendarName,
    Date startTime)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## getAllCalendarNames() Method

```
public String[] getAllCalendarNames()
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## getAllCalendars() Method

```
public CalendarDocument[] getAllCalendars()
    throws BusinessCalendarException, RemoteException
```

Gets all calendars, including rules info and mapped groups and users.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### getCalendar(String) Method

```
public CalendarDocument getCalendar(String calendarName)
    throws BusinessCalendarException, RemoteException
```

### Returns

*The CalendarDocument or null if a calendar with the given name does not exist.*

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### getGroupCalendar(String) Method

```
public CalendarDocument getGroupCalendar(String groupId)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### getGroupCalendarWalkHierarchy(String) Method

```
public CalendarDocument getGroupCalendarWalkHierarchy(String groupId)
    throws BusinessCalendarException, RemoteException
```

If a calendar is mapped to the group then that calendar is returned, otherwise returns the system calendar.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### getNextFreeMinute(CalendarDocument, Date) Method

```
public Date getNextFreeMinute(CalendarDocument cDoc,
    Date time)
```

throws `BusinessCalendarException`, `RemoteException`

Returns the next free minute given the rules in the calendar passed from the time passed.

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## getSystemCalendar() Method

```
public CalendarDocument getSystemCalendar()  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## getUserCalendar(String) Method

```
public CalendarDocument getUserCalendar(String userId)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## getUserCalendarWalkHierarchy(String) Method

```
public CalendarDocument getUserCalendarWalkHierarchy(String userId)  
    throws BusinessCalendarException, RemoteException
```

If a calendar is mapped to the user then that calendar is returned, otherwise returns the system calendar. Note that the existence of a system calendar is ensured by the system so this method always returns a calendar.

### Exceptions

*BusinessCalendarException*

*RemoteException*

---

## insertUserCalendarMapping(String, String) Method

```
public void insertUserCalendarMapping(String userid,  
    String calendarId)  
    throws BusinessCalendarException, RemoteException
```

Same as updateUserCalendar.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### isTimeFree(CalendarDocument, Date) Method

```
public boolean isTimeFree(CalendarDocument cDoc,
                          Date time)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### isUserTimeFree(String, Date) Method

```
public boolean isUserTimeFree(String userId,
                              Date time)
    throws BusinessCalendarException, RemoteException
```

Looks up the user's calendar using getUserCalendarWalkHierarchy and checks if the time passed is free based on the rules in the calendar.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### setGroupCalendar(String, String) Method

```
public void setGroupCalendar(String groupId,
                             String calendarName)
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

### setUserCalendar(String, String) Method

```
public void setUserCalendar(String userId,
                             String calendarName)
    throws BusinessCalendarException, RemoteException
```

Set the user's calendar. Does not do anything is either the given user or the calendar does not exist. A user can have only one calendar, so if a calendar is already mapped to the user then it is unmapped.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTime(String, String, Date) Method

```
public Date subtractBusinessTime(String interval,  
                                String userOrGroupId,  
                                Date startTime)  
    throws BusinessCalendarException, RemoteException
```

Go back in time by the given interval.

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTime(String, String) Method

```
public Date subtractBusinessTime(String interval,  
                                String userOrGroupId)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTime(String, CalendarDocument) Method

```
public Date subtractBusinessTime(String interval,  
                                CalendarDocument cDoc)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTime(String, CalendarDocument, Date) Method

```
public Date subtractBusinessTime(String interval,  
                                CalendarDocument cDoc,
```

`Date startTime)`  
`throws BusinessCalendarException, RemoteException`

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTimeCalendarName(String, String) Method

```
public Date subtractBusinessTimeCalendarName(String interval,  
                                             String calendarName)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTimeCalendarName(String, String, Date) Method

```
public Date subtractBusinessTimeCalendarName(String interval,  
                                             String calendarName,  
                                             Date startTime)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## subtractBusinessTimeSystemCalendar(String, Date) Method

```
public Date subtractBusinessTimeSystemCalendar(String interval,  
                                              Date startTime)  
    throws BusinessCalendarException, RemoteException
```

### Exceptions

*BusinessCalendarException*  
*RemoteException*

---

## systemCalendarExists() Method

```
public boolean systemCalendarExists()  
    throws BusinessCalendarException, RemoteException
```

**Exceptions**

*BusinessCalendarException*  
*RemoteException*

---

**unsetGroupCalendar(String) Method**

```
public void unsetGroupCalendar(String groupId)
    throws BusinessCalendarException, RemoteException
```

**Exceptions**

*BusinessCalendarException*  
*RemoteException*

---

**unsetUserCalendar(String) Method**

```
public void unsetUserCalendar(String userId)
    throws BusinessCalendarException, RemoteException
```

Unset the user's calendar.

**Exceptions**

*BusinessCalendarException*  
*RemoteException*

---

**updateCalendar(String, CalendarDocument) Method**

```
public void updateCalendar(String oldCalendarName,
    CalendarDocument cDoc)
    throws BusinessCalendarException, RemoteException
```

Update the calendar – oldCalendarName, using the passed CalendarDocument. The name of the calendar is also updated based on the name passed in. Use updateSystemCalendar to update the system calendar.

**Exceptions**

*BusinessCalendarException*  
 if the CalendarDocument indicates that this is a system calendar or if the system calendar has the same name as this calendar or if the name is being updated and a calendar already exists with that name.  
*RemoteException*

---



**updateSystemCalendar(CalendarDocument) Method**

```
public void updateSystemCalendar(CalendarDocument cDoc)  
    throws BusinessException, RemoteException
```

Update the system calendar using the passed CalendarDocument. If the name of the calendar is different than the name of the existing system calendar, then the existing system calendar is changed to non-system calendar. If a calendar with this name does not exist then it is created.

**Exceptions**

*BusinessCalendarException*

*RemoteException*

# BusinessCalendarException Class

*public class **BusinessCalendarException***

extends Exception

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      BusinessCalendarException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

***BusinessCalendarException**(String message, Throwable cause)*

***BusinessCalendarException**(String message)*

## **Method Summary**

public  
String **getMessage()**

### **Methods from java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getStackTrace,  
initCause, printStackTrace, printStackTrace, printStackTrace,  
setStackTrace, toString

### **Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### BusinessCalendarException

```
public BusinessCalendarException(String message,  
                                Throwable cause)
```

---

### BusinessCalendarException

```
public BusinessCalendarException(String message)
```

## Method Detail

### getMessage() Method

```
public String getMessage()
```

#### *Overrides*

```
    Throwable.getMessage()
```

# BusinessCalendarFactory Class

*public class **BusinessCalendarFactory***

extends `Object`

This factory is used to create an instance of `BusinessCalendar`.

---

## *Hierarchy*

`Object`

`BusinessCalendarFactory`

---

## Constructor Summary

***BusinessCalendarFactory()***

## Method Summary

`public static  
BusinessCalendar create()`

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait`

## Constructor Detail

### **BusinessCalendarFactory**

`public BusinessCalendarFactory()`

## Method Detail

## **create() Method**

```
public static BusinessCalendar create()  
    throws BusinessCalendarException
```

### **Returns**

*BusinessCalendar*

### **Exceptions**

*BusinessCalendarException*

# BusinessCalendarHome Interface

*public interface* **BusinessCalendarHome**

extends EJBHome

---

## *All Superinterfaces*

EJBHome, Remote

---

## Method Summary

public  
BusinessCalendar **create()**

## Methods from interface javax.ejb.EJBHome

getEJBMetaData, getHomeHandle, remove, remove

## Method Detail

### create() Method

```
public BusinessCalendar create()  
    throws RemoteException, CreateException
```

### Exceptions

*RemoteException*  
*CreateException*

# **com.bea.wli.management.archiving Package**

## **Interface Summary**

DocumentDataType These constants represent types of data stored in WLI\_DOCUMENT\_DATA.DATA.

TrackingEventType

## **Class Summary**

ArchivingPolicy

## **Exception Summary**

WLIArchiverException

# ArchivingPolicy Class

*public class ArchivingPolicy*

extends Object  
implements Serializable

---

## Hierarchy

Object  
ArchivingPolicy

## All Implemented Interfaces

Serializable

---

## Field Summary

public static  
final ArchivingPolicy **ARCHIVE\_OFF**  
Disable archiving of data for a process type

public static  
final ArchivingPolicy **ARCHIVE\_ON**  
Enable archiving of data for a process type

public static  
final ArchivingPolicy **DEFAULT**  
Use a global default when determining if data a given WF needs to be archived.

public static  
final List **VALUES**  
List of all ArchivingPolicy enumerated types.

## Method Summary

public int  
    **intValue()**  
Returns an integer representation for this ArchivingPolicy

public static  
ArchivingPolicy **toArchivingPolicy(int val)**  
Returns the ArchivingPolicy object whose integer representation is the given integer value.

public static  
ArchivingPolicy **toArchivingPolicy(String s)**



Returns the ArchivingPolicy object whose String representation is the given String value.

```
public String  
    toString()  
    Returns string representation of this ArchivingPolicy
```

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### ARCHIVE\_OFF

```
public static final ArchivingPolicy ARCHIVE_OFF
```

Disable archiving of data for a process type

---

### ARCHIVE\_ON

```
public static final ArchivingPolicy ARCHIVE_ON
```

Enable archiving of data for a process type

---

### DEFAULT

```
public static final ArchivingPolicy DEFAULT
```

Use a global default when determining if data a given WF needs to be archived.

---

### VALUES

```
public static final List VALUES
```

List of all ArchivingPolicy enumerated types.

## Method Detail

### intValue() Method

```
public int intValue()
```

Returns an integer representation for this ArchivingPolicy

### Related Topics

`ArchivingPolicy.toArchivingPolicy(int)`

---

### toArchivingPolicy(int) Method

```
public static ArchivingPolicy toArchivingPolicy(int val)
```

Returns the ArchivingPolicy object whose integer representation is the given integer value.

---

### toArchivingPolicy(String) Method

```
public static ArchivingPolicy toArchivingPolicy(String s)
```

Returns the ArchivingPolicy object whose String representation is the given String value.

---

### toString() Method

```
public String toString()
```

Returns string representation of this ArchivingPolicy

### Overrides

`Object.toString()`

# DocumentDataType Interface

*public interface DocumentDataType*

These constants represent types of data stored in WLI\_DOCUMENT\_DATA.DATA. These types apply to archived data. All of these are stored in BLOB column, however depending on the type, can be interpreted as text.

## Field Summary

```

    public
    static TYPE_EXCEPTION
    final int
           Text representation of a java.lang.Exception.

    public
    static TYPE_OTHER
    final int
           int

    public
    static TYPE_RAW_DATA
    final int
           Contents of some com.bea.data.RawData.

    public
    static TYPE_STRING
    final int
           Arbitrary string.

    public
    static TYPE_XML_TEXT
    final int
           XML text

```

## Field Detail

### TYPE\_EXCEPTION

```
public static final int TYPE_EXCEPTION
```

Text representation of a java.lang.Exception.

---

### TYPE\_OTHER

```
public static final int TYPE_OTHER
```

---

### TYPE\_RAW\_DATA

```
public static final int TYPE_RAW_DATA
```

Contents of some `com.bea.data.RawData`.

---

## **TYPE\_STRING**

```
public static final int TYPE_STRING
```

Arbitrary string.

---

## **TYPE\_XML\_TEXT**

```
public static final int TYPE_XML_TEXT
```

XML text

# TrackingEventType Interface

*public interface TrackingEventType*

## Field Summary

```

    public
    static EVENT_TYPE_CHILD_PROCESS_INSTANCE_CONTINUE
    final int
        Contains process_instance ID of the child process called via ProcessControl

    public
    static EVENT_TYPE_CHILD_PROCESS_INSTANCE_START
    final int
        Contains process_instance ID of the child process started via ProcessControl

    public
    static EVENT_TYPE_PARENT_PROCESS_INSTANCE_CONTINUE
    final int
        Contains process_instance ID of the parent process that called this process via
        ProcessControl

    public
    static EVENT_TYPE_PARENT_PROCESS_INSTANCE_START
    final int
        Contains process_instance ID of the parent process that started this process via
        ProcessControl

    public
    static EVENT_TYPE_PROCESS_ABORTED
    final int
        Indicates when an instance of the process is aborted due to an unhandled exception

    public
    static EVENT_TYPE_PROCESS_ACTIVITY_ABORT
    final int
        Indicates the abnormal end of a process node execution.

    public
    static EVENT_TYPE_PROCESS_ACTIVITY_END
    final int
        Indicates the normal end of a process node execution.

    public
    static EVENT_TYPE_PROCESS_ACTIVITY_START
    final int
        Indicates the start of a process node execution.

    public
    static EVENT_TYPE_PROCESS_FROZEN
    final int
        Process was frozen due to errors but can be restarted

    public
    static EVENT_TYPE_PROCESS_LABEL
    final int
        Indicates a change in the process label.

    public
    static EVENT_TYPE_PROCESS_LOG
    final int
        Indicates when a user invokes JpdContext.log()

    public
    static EVENT_TYPE_PROCESS_RESUMED
    final int

```

Indicates that the process was resumed by administrator.

```
public
static EVENT_TYPE_PROCESS_SUSPENDED
final int
```

Indicates that the process was suspended by administrator.

```
public
static EVENT_TYPE_PROCESS_TERMINATED
final int
```

The process was manually terminated from OAMconsole.

```
public
static EVENT_TYPE_PROCESS_UNFROZEN
final int
```

A frozen process was restarted

```
public
static EVENT_TYPE_UNHANDLED_EXCEPTION
final int
```

Indicates that an exception was raised during process execution and this exception was not handled by any exception handler.

## Field Detail

### **EVENT\_TYPE\_CHILD\_PROCESS\_INSTANCE\_CONTINUE**

```
public static final int EVENT_TYPE_CHILD_PROCESS_INSTANCE_CONTINUE
```

Contains process\_instance ID of the child process called via ProcessControl

---

### **EVENT\_TYPE\_CHILD\_PROCESS\_INSTANCE\_START**

```
public static final int EVENT_TYPE_CHILD_PROCESS_INSTANCE_START
```

Contains process\_instance ID of the child process started via ProcessControl

---

### **EVENT\_TYPE\_PARENT\_PROCESS\_INSTANCE\_CONTINUE**

```
public static final int EVENT_TYPE_PARENT_PROCESS_INSTANCE_CONTINUE
```

Contains process\_instance ID of the parent process that called this process via ProcessControl

---

### **EVENT\_TYPE\_PARENT\_PROCESS\_INSTANCE\_START**

```
public static final int EVENT_TYPE_PARENT_PROCESS_INSTANCE_START
```

Contains process\_instance ID of the parent process that started this process via ProcessControl

---

**EVENT\_TYPE\_PROCESS\_ABORTED**

```
public static final int EVENT_TYPE_PROCESS_ABORTED
```

Indicates when an instance of the process is aborted due to an unhandled exception

---

**EVENT\_TYPE\_PROCESS\_ACTIVITY\_ABORT**

```
public static final int EVENT_TYPE_PROCESS_ACTIVITY_ABORT
```

Indicates the abnormal end of a process node execution. For example, 2 branches may be started for a choice. One is executed, the other branch is aborted.

---

**EVENT\_TYPE\_PROCESS\_ACTIVITY\_END**

```
public static final int EVENT_TYPE_PROCESS_ACTIVITY_END
```

Indicates the normal end of a process node execution.

---

**EVENT\_TYPE\_PROCESS\_ACTIVITY\_START**

```
public static final int EVENT_TYPE_PROCESS_ACTIVITY_START
```

Indicates the start of a process node execution.

---

**EVENT\_TYPE\_PROCESS\_FROZEN**

```
public static final int EVENT_TYPE_PROCESS_FROZEN
```

Process was frozen due to errors but can be restarted

---

**EVENT\_TYPE\_PROCESS\_LABEL**

```
public static final int EVENT_TYPE_PROCESS_LABEL
```

Indicates a change in the process label.

---

**EVENT\_TYPE\_PROCESS\_LOG**

```
public static final int EVENT_TYPE_PROCESS_LOG
```

Indicates when a user invokes JpdContext.log()

---

**EVENT\_TYPE\_PROCESS\_RESUMED**

```
public static final int EVENT_TYPE_PROCESS_RESUMED
```

Indicates that the process was resumed by administrator.

---

**EVENT\_TYPE\_PROCESS\_SUSPENDED**

```
public static final int EVENT_TYPE_PROCESS_SUSPENDED
```

Indicates that the process was suspended by administrator.

---

**EVENT\_TYPE\_PROCESS\_TERMINATED**

```
public static final int EVENT_TYPE_PROCESS_TERMINATED
```

The process was manually terminated from OAMconsole.

---

**EVENT\_TYPE\_PROCESS\_UNFROZEN**

```
public static final int EVENT_TYPE_PROCESS_UNFROZEN
```

A frozen process was restarted

---

**EVENT\_TYPE\_UNHANDLED\_EXCEPTION**

```
public static final int EVENT_TYPE_UNHANDLED_EXCEPTION
```

Indicates that an exception was raised during process execution and this exception was not handled by any exception handler. This event includes the exception.



# WLIArchiverException Class

*public class WLIArchiverException*

extends Exception

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      WLIArchiverException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

*WLIArchiverException(String msg, Throwable rootCause)*

*WLIArchiverException(String msg)*

## **Method Summary**

### **Methods from java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getMessage,  
getStackTrace, initCause, printStackTrace, printStackTrace,  
printStackTrace, setStackTrace, toString

### **Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### WLIArchiverException

```
public WLIArchiverException(String msg,  
                             Throwable rootCause)
```

---

### WLIArchiverException

```
public WLIArchiverException(String msg)
```

# com.bea.wli.management.configuration Package

## Interface Summary

B2BBaseMBean	
B2BConfigMBean	The B2B interface represents the WebLogic B2B Integration.
BPMConfigurationMBean	MBean for managing general properties of BPM subsystem.
BusinessCalendarConfigurationMBean	MBean for configuring business calendars.
EmailEventGenConfigurationMBean	Manages channel rules for a Email Event Generator.
EmailEventGeneratorsMBean	Provides a central access point for getting/creating Email Event Generator MBeans
FileEventGenConfigurationMBean	Manages channel rules for a File Event Generator.
FileEventGeneratorsMBean	Provides a central access point for getting/creating File Event Generator MBeans
GenericEventGenConfigurationMBean	Manages channel rules for a Generic Event Generator.
GenericEventGeneratorsMBean	Provides a central access point for getting/creating Event Generator MBeans
JMSEventGenConfigurationMBean	Manages channel rules for a JMS Event Generator.
JMSEventGeneratorsMBean	Provides a central access point for getting/creating JMS Event Generator MBeans
MessageBrokerConfigurationMBean	
PasswordStoreConfigurationMBean	Manages the WLI password store
ProcessConfigurationMBean	MBean for managing properties of a particular <b>type</b> of process.
ProcessControlMBean	MBean for managing properties of a Process Control
ProcessNodeInfo	Represents a node in the process definition
SampleConfigurationMBean	
ServiceBrokerControlMBean	MBean for managing properties of a Service Broker Control
TimerEventGenConfigurationMBean	Manages channel rules for a Timer Event Generator.
TimerEventGeneratorsMBean	Provides a central access point for getting/creating Timer Event Generator MBeans
UserInfoConfigurationMBean	
VersioningConfigurationMBean	Configuration MBean interface for Users in WLI.
WLIconsoleEventGenConfigurationMBean	Configuration MBean interface for Processes' Versioning.
WorklistConfigurationMBean	Configuration MBean interface for WLI Console Extension for External Event Generators There is one instance of WLIconsoleEventGenConfigurationMBean per each type of generator.
	MBean for configuring the worklist

## Class Summary

EmailEventGenChannelConfiguration	Represents a channel rule for an EMail Event Generator.
-----------------------------------	---

## Java Class Reference

FileEventGenChannelConfiguration	Represents a channel rule for a File Event Generator.
GenericEventGenChannelConfiguration	
JMSEventGenChannelConfiguration	Represents a channel rule for a JMS Event Generator.
NodeMethodInfo	Entry into map from a process's method name to information about its method.
ProcessVariable	Represents a process variable with its name and type (class)
TimerEventGenChannelConfiguration	Represents a channel rule for a Timer Event Generator.
TrackingLevel	Typesafe enumerated class for describing tracking levels for processes.

## Exception Summary

B2BConfigurationException	A B2BConfigurationException is thrown if an error occurred in a WebLogic B2B Integration management application.
---------------------------	--

# B2BBaseMBean Interface

*public interface* **B2BBaseMBean**

extends ConfigurationMBean

---

## *All Superinterfaces*

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

## *All Known Subinterfaces*

B2BConfigMBean

---

## Field Summary

Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public
Object clone(String name)
    Makes a deep copy of the Business Protocol with the name passed.

public
void delete()
    Delete the object

public
void fromXML(String xmlString)
    Take XML representation of a child MBean and its children

public
Object getDelegateObject()

public
void setDelegateObject(Object o)

public
String toXML()
    XML representation for the current and its subchild

public
void update()
    Updates the .
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`,  
`getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`,  
`isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`,  
`restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`,  
`setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`,  
`setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`,  
`isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Method Detail

### `clone(String)` Method

```
public Object clone(String name)
    throws B2BConfigurationException
```

Makes a deep copy of the Business Protocol with the name passed.

#### Parameters

*name*  
 name for the new Business Protocol.

#### Returns

*java.lang.Object* – The new object.

#### Exceptions

*B2BConfigurationException*

---

## delete() Method

```
public void delete()  
    throws B2BConfigurationException
```

Delete the object

### Exceptions

*B2BConfigurationException*

---

## fromXML(String) Method

```
public void fromXML(String xmlString)  
    throws B2BConfigurationException
```

Take XML representation of a child MBean and its children

### Exceptions

*B2BConfigurationException*

---

## getDelegateObject() Method

```
public Object getDelegateObject()  
    throws B2BConfigurationException
```

### Exceptions

*B2BConfigurationException*

---

## setDelegateObject(Object) Method

```
public void setDelegateObject(Object o)
```

---

## toXML() Method

```
public String toXML()  
    throws B2BConfigurationException
```

XML representation for the current and its subchild

### Exceptions

*B2BConfigurationException*

---

## **update() Method**

```
public void update()  
    throws B2BConfigurationException
```

Updates the .

### **Exceptions**

*B2BConfigurationException*

---



## B2BConfigMBean Interface

*public interface **B2BConfigMBean***

extends B2BBaseMBean

The B2B interface represents the WebLogic B2B Integration. It acts as a factory for all other mbeans.

---

### *All Superinterfaces*

B2BBaseMBean, ConfigurationMBean, DynamicMBean, MBeanRegistration, NotificationBroadcaster, WebLogicMBean

---

## Field Summary

Fields from interface `weblogic.management.configuration.ConfigurationMBean`

CACHING\_STUB\_SVUID

## Method Summary

```
public
String getAuditLogClass()
    Gets the audit log class name.

public
String getCertificateVerificationClass()
    Gets the certification verification class name.

public
String getProxyHost()
    Gets the proxy host of the WebLogic B2B Integration.

public
String getProxyPort()
    Gets the proxy port of the WebLogic B2B Integration.

public
String getSecureTimestampClass()
    Gets the secure timestamp class name.

public
void refreshKeyStores()
    Refresh Keystores

public
void setAuditLogClass(String name)
    Sets the audit log class name.
```

```

public setCertificateVerificationClass(String name)
    void          Sets the certificate verification class name.
public
    void setProxyHost(String host)
                Sets the proxy host of the WebLogic B2B Integration.
public
    void setProxyPort(String port)
                Sets the proxy port of the WebLogic B2B Integration.
public
    void setSecureTimestampClass(String name)
                Sets the secure timestamp class name.
public
    void update()
                Update the configuration for WebLogic B2B Integration server.
public
    void updateAll(String proxyHost, String proxyPort, String certificateVerificationClass,
                  String auditLogClass, String secureTimestampClass)
                Update all the configuration for WebLogic B2B Integration server.

```

**Methods from interface com.bea.wli.management.configuration.B2BBaseMBean**

clone, delete, fromXML, getDelegateObject, setDelegateObject, toXML,

**Methods from interface weblogic.management.configuration.ConfigurationMBean**

freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

**Methods from interface javax.management.DynamicMBean**

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

**Methods from interface javax.management.MBeanRegistration**

postDeregister, postRegister, preDeregister, preRegister

**Methods from interface javax.management.NotificationBroadcaster**

addNotificationListener, getNotificationInfo, removeNotificationListener

**Methods from interface weblogic.management.WebLogicMBean**

getMBeanInfo, getName, getObjectName, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Method Detail

### getAuditLogClass() Method

```
public String getAuditLogClass()
```

Gets the audit log class name.

#### Returns

*String* – audit log class name.

---

### getCertificateVerificationClass() Method

```
public String getCertificateVerificationClass()
```

Gets the certification verification class name.

#### Returns

*String* – certificate verification class name.

---

### getProxyHost() Method

```
public String getProxyHost()
```

Gets the proxy host of the WebLogic B2B Integration.

#### Returns

*String* – proxy host of the WebLogic B2B Integration.

---

### getProxyPort() Method

```
public String getProxyPort()
```

Gets the proxy port of the WebLogic B2B Integration.

#### Returns

*String* – proxy port of the WebLogic B2B Integration.

---

## getSecureTimestampClass() Method

```
public String getSecureTimestampClass()
```

Gets the secure timestamp class name.

### Returns

*String* – secure timestamp class name.

---

## refreshKeyStores() Method

```
public void refreshKeyStores()
```

Refresh Keystores

---

## setAuditLogClass(String) Method

```
public void setAuditLogClass(String name)
```

Sets the audit log class name.

### Parameters

*name*

audit log class name.

---

## setCertificateVerificationClass(String) Method

```
public void setCertificateVerificationClass(String name)
```

Sets the certificate verification class name.

### Parameters

*name*

certificate verification class name.

---

## setProxyHost(String) Method

```
public void setProxyHost(String host)
```

Sets the proxy host of the WebLogic B2B Integration.

## Parameters

*host*

proxy host for the WebLogic B2B Integration.

---

## setProxyPort(String) Method

```
public void setProxyPort(String port)
```

Sets the proxy port of the WebLogic B2B Integration.

## Parameters

*port*

proxy port for the WebLogic B2B Integration.

---

## setSecureTimestampClass(String) Method

```
public void setSecureTimestampClass(String name)
```

Sets the secure timestamp class name.

## Parameters

*name*

secure timestamp class name.

---

## update() Method

```
public void update()  
    throws B2BConfigurationException
```

Update the configuration for WebLogic B2B Integration server.

## Exceptions

*B2BConfigurationException*

If an error occurred during update.

---

## updateAll(String, String, String, String, String) Method

```
public void updateAll(String proxyHost,  
                     String proxyPort,  
                     String certificateVerificationClass,  
                     String auditLogClass,  
                     String secureTimestampClass)  
    throws B2BConfigurationException
```

Update all the configuration for WebLogic B2B Integration server.

### Exceptions

*B2BConfigurationException*

If an error occurred during update.

# B2BConfigurationException Class

*public class B2BConfigurationException*

extends `Exception`

A `B2BConfigurationException` is thrown if an error occurred in a WebLogic B2B Integration management application.

---

## *Hierarchy*

```
Object
  Throwable
    Exception
      B2BConfigurationException
```

## *All Implemented Interfaces*

`Serializable`

---

## Constructor Summary

### *B2BConfigurationException()*

Constructs a `B2BConfigurationException`.

### *B2BConfigurationException(Exception e)*

Constructs a `B2BConfigurationException` with a nested exception

### *B2BConfigurationException(String msg)*

Constructs a `B2BConfigurationException` with the specified message text.

## Method Summary

### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getMessage`, `getStackTrace`, `initCause`, `printStackTrace`, `printStackTrace`,

`printStackTrace, setStackTrace, toString`

#### Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait`

## Constructor Detail

### **B2BConfigurationException**

```
public B2BConfigurationException()
```

Constructs a B2BConfigurationException.

---

### **B2BConfigurationException**

```
public B2BConfigurationException(Exception e)
```

Constructs a B2BConfigurationException with a nested exception

---

### **B2BConfigurationException**

```
public B2BConfigurationException(String msg)
```

Constructs a B2BConfigurationException with the specified message text.

## Method Detail



# BPMConfigurationMBean Interface

*public interface* **BPMConfigurationMBean**

extends ConfigurationMBean

MBean for managing general properties of BPM subsystem. The properties exposed via this MBean can be changed to affect certain aspects of BPM engine.

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public long
    getArchiveProcessInterval()
        Returns time in milliseconds to indicate how often Archive process is run.

public long
    getArchiveProcessStart()
        UTC time in milliseconds when the 1st automated invocation of the Archive
        process will take place.

public String
    getArchiverDataSourceName()
        Returns JNDI name for the DataSource that points to archvie DB.

public boolean
    getArchiverEnabled()
        Returns whether the archiver is enabled.

public
ArchivingPolicy getDefaultArchivingPolicy()
        Returns the default archiving policy for processes whose archiving policy is
```

## Java Class Reference

ArchivingPolicy.DEFAULT.

```
public
TrackingLevel getDefaultProcessTrackingLevel()
    Returns the default Process tracking level for processes whose tracking level
    is TrackingLevel.DEFAULT.

public int
getPurgeDelay()
    Returns the purge delay in seconds.

public void
registerAdminMBeans(byte[] appManifest, String appName)
    Internal operation that is used by managed servers to trigger MBean
    registration upon deployment.

public void
registerVersioningMBeans(Map configMap, String appName)
    Internal operation that is used by managed servers to trigger
    Versioning-specific MBean registration upon application deployment.

public void
setArchiveProcessInterval(long time)
    Sets time in milliseconds to indicate how often Archive process is run.

public void
setArchiveProcessStart(long time)
    Sets UTC time in milliseconds when the 1st automated invocation of the
    Archive process will take place.

public void
setArchiverDataSourceName(String jndiName)
    Sets JNDI name for the DataSource that points to archive DB.

public void
setArchiverEnabled(boolean isEnabled)
    Global switch to enable/disable WLI Archiver.

public void
setDefaultArchivingPolicy(ArchivingPolicy newArchivingPolicy)
    Sets the default archiving policy for processes whose archiving policy is
    ArchivingPolicy.DEFAULT.

public void
setDefaultProcessTrackingLevel(TrackingLevel newTrackingLevel)
    Sets the default process tracking level for processes whose tracking level is
    TrackingLevel.DEFAULT.

public void
setPurgeDelay(int delay)
    Sets the purge delay in seconds.

public void
unregisterAdminMBeans(byte[] appManifest, String appName)
    Internal operation that is used by managed servers to trigger.

public void
unregisterVersioningMBeans(Map configMap, String appName)
    Internal operation that is used by managed servers to trigger
    Versioning-specific MBean unregistration upon application deployment.
```

```
public void updateArchiveProcessStartPropertyOnly(long time)
```

#### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

```
freezeCurrentValue, getAttributeStringValue, getComments, getNotes,  
getSetFields, getXml, getXmlConverter, isDefaultedMBean,  
isPersistenceEnabled, preDeregister, registerConfigMBean,  
restoreDefaultValue, setComments, setDefaultedMBean, setNotes,  
setPersistenceEnabled, touch, unregisterConfigMBean
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,  
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectNames, getParent, getType,  
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### getArchiveProcessInterval() Method

```
public long getArchiveProcessInterval()
```

Returns time in milliseconds to indicate how often Archive process is run.

---

### getArchiveProcessStart() Method

```
public long getArchiveProcessStart()
```

UTC time in milliseconds when the 1st automated invocation of the Archive process will take place.

#### Returns

*UTC time in milliseconds when the 1st automated invocation of the Archive process will take place.*

---

### getArchiverDataSourceName() Method

```
public String getArchiverDataSourceName()
```

Returns JNDI name for the DataSource that points to archvie DB.

#### Returns

*JNDI name for the DataSource that points to archvie DB.*

---

### getArchiverEnabled() Method

```
public boolean getArchiverEnabled()
```

Returns whether the archiver is enabled.

#### Returns

*whether the archiver is enabled.*

---

### getDefaultArchivingPolicy() Method

```
public ArchivingPolicy getDefaultArchivingPolicy()
```

Returns the default archiving policy for processes whose archiving policy is ArchivingPolicy.DEFAULT.

**Returns**

*the default archiving policy for processes whose archiving policy is `ArchivingPolicy.DEFAULT`.*

---

**getDefaultProcessTrackingLevel() Method**

```
public TrackingLevel getDefaultProcessTrackingLevel()
```

Returns the default Process tracking level for processes whose tracking level is `TrackingLevel.DEFAULT`.

**Returns**

*the default Process tracking level for processes whose tracking level is `TrackingLevel.DEFAULT`.*

---

**getPurgeDelay() Method**

```
public int getPurgeDelay()
```

Returns the purge delay in **seconds**.

**Returns**

*purge delay in **seconds**.*

---

**registerAdminMBeans(byte[], String) Method**

```
public void registerAdminMBeans(byte[] appManifest,
                                String appName)
    throws MBeanException
```

Internal operation that is used by managed servers to trigger MBean registration upon deployment.

**Parameters**

*appManifest*  
the application Manifest

*appName*  
the application Name

**Exceptions**

*MBeanException*  
on error

---

## registerVersioningMBeans(Map, String) Method

```
public void registerVersioningMBeans(Map configMap,  
                                     String appName)  
    throws MBeanException
```

Internal operation that is used by managed servers to trigger Versioning-specific MBean registration upon application deployment. This is called for packaged EARS and exploded EARS as well

### Parameters

*configMap*  
configuration map

*appName*  
the application name

### Exceptions

*MBeanException*  
on error

---

## setArchiveProcessInterval(long) Method

```
public void setArchiveProcessInterval(long time)  
    throws MBeanException
```

Sets time in milliseconds to indicate how often Archive process is run.

### Parameters

*time*  
the archive interval in milliseconds

### Exceptions

*MBeanException*

---

## setArchiveProcessStart(long) Method

```
public void setArchiveProcessStart(long time)  
    throws MBeanException
```

Sets UTC time in milliseconds when the 1st automated invocation of the Archive process will take place.

### Parameters

*time*  
archive process time in UTC milliseconds.

## Exceptions

*MBeanException*

---

### setArchiverDataSourceName(String) Method

```
public void setArchiverDataSourceName(String jndiName)
        throws MBeanException
```

Sets JNDI name for the DataSource that points to archive DB.

#### Parameters

*jndiName*

The JNDI name of the data source where the archive data is stored.

#### Exceptions

*MBeanException*

---

### setArchiverEnabled(boolean) Method

```
public void setArchiverEnabled(boolean isEnabled)
        throws MBeanException
```

Global switch to enable/disable WLI Archiver.

#### Parameters

*isEnabled*

true to enable the archiver, false to disable.

#### Exceptions

*MBeanException*

---

### setDefaultArchivingPolicy(ArchivingPolicy) Method

```
public void setDefaultArchivingPolicy(ArchivingPolicy newArchivingPolicy)
        throws MBeanException, InvalidAttributeValueException
```

Sets the default archiving policy for processes whose archiving policy is `ArchivingPolicy.DEFAULT`.

#### Parameters

*newArchivingPolicy*

the new archiving policy

**Exceptions***MBeanException*

if another error occurs while setting the archiving policy

*InvalidAttributeValueException*

if an invalid policy is provided

**setDefaultProcessTrackingLevel(TrackingLevel) Method**

```
public void setDefaultProcessTrackingLevel(TrackingLevel newTrackingLevel)
    throws MBeanException, InvalidAttributeValueException
```

Sets the default process tracking level for processes whose tracking level is `TrackingLevel.DEFAULT`.

**Parameters***newTrackingLevel*

the new process tracking level

**Exceptions***MBeanException*

indicates any other exception

*InvalidAttributeValueException*when `newTrackingLevel` is an invalid tracking level.**setPurgeDelay(int) Method**

```
public void setPurgeDelay(int delay)
    throws MBeanException
```

Sets the purge delay in **seconds**.

**Parameters***delay*the purge delay in **seconds**.**Exceptions***MBeanException***unregisterAdminMBeans(byte[], String) Method**

```
public void unregisterAdminMBeans(byte[] appManifest,
    String appName)
    throws MBeanException
```



Internal operation that is used by managed servers to trigger. MBean unregistration upon undeployment.

### Parameters

*appManifest*  
the application Manifest

*appName*  
the application Name

### Exceptions

*MBeanException*  
on error

---

## unregisterVersioningMBeans(Map, String) Method

```
public void unregisterVersioningMBeans(Map configMap,  
                                       String appName)  
    throws MBeanException
```

Internal operation that is used by managed servers to trigger Versioning-specific MBean unregistration upon application deployment. This is called for packaged EARS and exploded EARS as well.

### Parameters

*configMap*  
configuration map

*appName*  
the application name

### Exceptions

*MBeanException*  
on error

---

## updateArchiveProcessStartPropertyOnly(long) Method

```
public void updateArchiveProcessStartPropertyOnly(long time)  
    throws MBeanException
```

### Exceptions

*MBeanException*

# BusinessCalendarConfigurationMBean Interface

*public interface BusinessCalendarConfigurationMBean*

extends ConfigurationMBean

MBean for configuring business calendars.

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public boolean
    calendarNameExists(String name)
        Checks if a calendar with the given name exists.

public void
    createCalendar(CalendarDocument cDoc)
        Creates a new calendar from the information passed in the
        CalendarDocument.

public void
    deleteCalendar(String calendarName)

public void
    deleteCalendars(String[] names)

public String[]
    getAllCalendarNames()

public
CalendarDocument[] getAllCalendars()
    Gets all calendars, includin rules info and mapped groups and users
```

## Java Class Reference

```
    public getCalendar(String calendarName)
CalendarDocument
    public
CalendarDocument getGroupCalendar(String groupId)
    public
CalendarDocument getSystemCalendar()
    public
CalendarDocument getUserCalendar(String userId)
    public void
        setGroupCalendar(String groupId, String calendarName)
    public void
        setUserCalendar(String userId, String calendarName)
        Set the user's calendar.
    public boolean
        systemCalendarExists()
    public void
        unsetGroupCalendar(String groupId)
    public void
        unsetUserCalendar(String userId)
    public void
        updateCalendar(String oldCalendarName, CalendarDocument cDoc)
        Update the calendar – oldCalendarName, using the passed
        CalendarDocument.
    public void
        updateSystemCalendar(CalendarDocument cDoc)
        Update the system calendar using the passed CalendarDocument.
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`,  
`getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`,  
`isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`,  
`restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`,  
`setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`,  
`setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

**Methods from interface `weblogic.management.WebLogicMBean`**

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`,  
`isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the `serialVersionUID` of the generated MBean Stub (`xxxMBean_Stub`). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### `calendarNameExists(String)` Method

```
public boolean calendarNameExists(String name)
    throws MBeanException
```

Checks if a calendar with the given name exists.

#### Exceptions

*MBeanException*

---

### `createCalendar(CalendarDocument)` Method

```
public void createCalendar(CalendarDocument cDoc)
    throws MBeanException
```

Creates a new calendar from the information passed in the `CalendarDocument`.

## Exceptions

*MBeanException*

– if a calendar with the given name exists or of the document is not valid.

---

## deleteCalendar(String) Method

```
public void deleteCalendar(String calendarName)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## deleteCalendars(String[]) Method

```
public void deleteCalendars(String[] names)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## getAllCalendarNames() Method

```
public String[] getAllCalendarNames()
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## getAllCalendars() Method

```
public CalendarDocument[] getAllCalendars()
    throws MBeanException
```

Gets all calendars, includin rules info and mapped groups and users

## Exceptions

*MBeanException*

---

## getCalendar(String) Method

```
public CalendarDocument getCalendar(String calendarName)
    throws MBeanException
```

### Returns

*The CalendarDocument or null if a calendar with the given name does not exist.*

### Exceptions

*MBeanException*

---

## getGroupCalendar(String) Method

```
public CalendarDocument getGroupCalendar(String groupId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## getSystemCalendar() Method

```
public CalendarDocument getSystemCalendar()
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## getUserCalendar(String) Method

```
public CalendarDocument getUserCalendar(String userId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setGroupCalendar(String, String) Method

```
public void setGroupCalendar(String groupId,
                             String calendarName)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### setUserCalendar(String, String) Method

```
public void setUserCalendar(String userId,  
                           String calendarName)  
    throws MBeanException
```

Set the user's calendar. Does not do anything is either the given user or the calendar does not exist. A user can have only one calendar, so if a calendar is already mapped to the user then it is unmapped.

## Exceptions

*MBeanException*

---

### systemCalendarExists() Method

```
public boolean systemCalendarExists()  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### unsetGroupCalendar(String) Method

```
public void unsetGroupCalendar(String groupId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### unsetUserCalendar(String) Method

```
public void unsetUserCalendar(String userId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## updateCalendar(String, CalendarDocument) Method

```
public void updateCalendar(String oldCalendarName,  
                           CalendarDocument cDoc)  
    throws MBeanException
```

Update the calendar – oldCalendarName, using the passed CalendarDocument. The name of the calendar is also updated based on the name passed in. Use updateSystemCalendar to update the system calendar.

### Exceptions

*MBeanException*

---

## updateSystemCalendar(CalendarDocument) Method

```
public void updateSystemCalendar(CalendarDocument cDoc)  
    throws MBeanException
```

Update the system calendar using the passed CalendarDocument. If the name of the calendar is different than the name of the existing system calendar, then the existing system calendar is changed to non-system calendar. If a calendar with this name does not exist then it is created.

### Exceptions

*MBeanException*



# EmailEventGenChannelConfiguration Class

*public class EmailEventGenChannelConfiguration*

extends Object

implements Serializable

Represents a channel rule for an EMail Event Generator.

Rule Fields:

- hostname – mail server host address
- portNumber – mail server port number
- protocol – mail server protocol (pop3, imap etc.)
- username – mail server username
- password – Password to the above account
- password–alias – Alias to password store
- pollingInterval – how often to poll the server, in secs
- read–limit – Num of emails to read and process every poll interval
- postReadAction – What to do with the email after processing – delete | archive | move
- attachments – What to do with the attachemnts – archive | ignore
- archive–directory – Directory to archive messages and attachments in
- error–directory – Directory to archive messges with errors messages and attachments in
- imap–move–folder – Folder to move email messages to after they are processes (IMAP only)

Other fields:

- channel – channel to send to if rules match
- comment – comment for this channel rule
- packageType – package type to send (xml, RawData, String)
- attachmentArchiveDir – What to do with attachments – archive | post
- attachmentAction – What to do with attachments – archive | post
- archiveFolder – name of the folder to move the message to

## Related Topics

EmailEventGenConfigurationMBean

---

## Hierarchy

Object

EmailEventGenChannelConfiguration

## All Implemented Interfaces

Serializable

---

## Field Summary

```

    public
    static _ATTACHMENT_ARCHIVE
final String      String

    public
    static _ATTACHMENT_IGNORE
final String      String

    public
    static _DEFAULT_POLLING_INTERVAL
final int         int

    public
    static _DEFAULT_PROTOCOL
final String      String

    public
    static _DEFAULT_READ_LIMIT
final int         int

    public
    static _POSTREAD_ARCHIVE
final String      String

    public
    static _POSTREAD_DELETE
final String      String

    public
    static _POSTREAD_MOVE
final String      String

```

## Constructor Summary

***EmailEventGenChannelConfiguration()***

Constructs a new EmailEventGenChannelConfiguration object.

## Method Summary

```

public
String getArchiveDirectory()
    Return the archive directory.

public
String getAttachmentAction()
    Return the attachment action.

```

```

public getChannel()
String      Return the channel name.

public
String getComment()
            Return the description.

public
String getErrorDirectory()
            Return the name of the error directory.

public
String getHostname()
            Return the host name.

public
String getImapMoveFolder()
            Return the archive folder.

public
String getPassword()
            Return the password.

public
String getPasswordAlias()
            Return the password alias.

public
    int getPollingInterval()
            Return the polling interval.

public
    int getPortNumber()
            Return the port number.

public
String getPostReadAction()
            Return the post read action.

public
String getProtocol()
            Return the protocol.

public
String getPublishAsUser()
            Get the publishAsUser attribute

public
    int getReadLimit()
            Return the read limit.

public
String getSubjectFilter()

public
String getUsername()
            Return the username.

public
    void setArchiveDirectory(String archiveDir)
            Set the archive directory.

```

```

public setAttachmentAction(String attachmentAction)
    void          Sets the attachment action.
public
    void setChannel(String channel)
        Set the name of the channel.
public
    void setComment(String comment)
        Set a description for the channel rule.
public
    void setErrorDirectory(String errorDir)
        Set the error directory.
public
    void setHostname(String hostName)
        Set the hostname.
public
    void setImapMoveFolder(String imapMoveFolder)
        Set the IMAP move folder name.
public
    void setPassword(String password)
        Set the password to the email account being polled.
public
    void setPasswordAlias(String pwdAlias)
        Set the password alias.
public
    void setPollingInterval(int pollingInterval)
        Set the polling interval.
public
    void setPortNumber(int portNumber)
        Set the port number on which the email server is accessible.
public
    void setPostReadAction(String postReadAction)
        Set the post-read action.
public
    void setProtocol(String protocol)
        Set the protocol to use to access the specified mailbox.
public
    void setPublishAsUser(String publishAsUser)
        Set the publishAsUser attribute
public
    void setReadLimit(int readLimit)
        Set the read limit.
public
    void setSubjectFilter(String subjectFilter)
public
    void setUsername(String username)
        Set the username of the email account that needs to be polled for incoming messages.

```

```
public toString()
String      Returns a string representation of this channel.
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### **\_ATTACHMENT\_ARCHIVE**

```
public static final String _ATTACHMENT_ARCHIVE
```

---

### **\_ATTACHMENT\_IGNORE**

```
public static final String _ATTACHMENT_IGNORE
```

---

### **\_DEFAULT\_POLLING\_INTERVAL**

```
public static final int _DEFAULT_POLLING_INTERVAL
```

---

### **\_DEFAULT\_PROTOCOL**

```
public static final String _DEFAULT_PROTOCOL
```

---

### **\_DEFAULT\_READ\_LIMIT**

```
public static final int _DEFAULT_READ_LIMIT
```

---

### **\_POSTREAD\_ARCHIVE**

```
public static final String _POSTREAD_ARCHIVE
```

---

### **\_POSTREAD\_DELETE**

```
public static final String _POSTREAD_DELETE
```

---

### **\_POSTREAD\_MOVE**

```
public static final String _POSTREAD_MOVE
```

## Constructor Detail

### EmailEventGenChannelConfiguration

```
public EmailEventGenChannelConfiguration()
```

Constructs a new EmailEventGenChannelConfiguration object.

## Method Detail

### getArchiveDirectory() Method

```
public String getArchiveDirectory()
```

Return the archive directory.

#### Returns

*the archive directory.*

---

### getAttachmentAction() Method

```
public String getAttachmentAction()
```

Return the attachment action.

#### Returns

*the value currently set for attachment action– ARCHIVE or IGNORE.*

---

### getChannel() Method

```
public String getChannel()
```

Return the channel name.

#### Returns

*the name of the channel.*

---

## getComment() Method

```
public String getComment()
```

Return the description.

---

## getErrorDirectory() Method

```
public String getErrorDirectory()
```

Return the name of the error directory.

### Returns

*the error directory.*

---

## getHostname() Method

```
public String getHostname()
```

Return the host name.

### Returns

*the host name.*

---

## getImapMoveFolder() Method

```
public String getImapMoveFolder()
```

Return the archive folder.

### Returns

*the name of the archive folder.*

---

## getPassword() Method

```
public String getPassword()
```

Return the password.

### Returns

*the password.*

---

## getPasswordAlias() Method

```
public String getPasswordAlias()
```

Return the password alias.

### Returns

*the password alias.*

---

## getPollingInterval() Method

```
public int getPollingInterval()
```

Return the polling interval.

### Returns

*the polling interval.*

---

## getPortNumber() Method

```
public int getPortNumber()
```

Return the port number.

### Returns

*the port number of the email server.*

---

## getPostReadAction() Method

```
public String getPostReadAction()
```

Return the post read action.

### Returns

*the value of the post read action.*

---

## getProtocol() Method

```
public String getProtocol()
```

Return the protocol.



**Returns**

*the name of the protocol currently configured.*

---

**getPublishAsUser() Method**

```
public String getPublishAsUser()
```

Get the publishAsUser attribute

**Returns**

*publishAsUser attribute*

---

**getReadLimit() Method**

```
public int getReadLimit()
```

Return the read limit.

**Returns**

*the read limit configured.*

---

**getSubjectFilter() Method**

**DEPRECATED** Return the subject filter

```
public String getSubjectFilter()
```

**Returns**

*the subject filter.*

---

**getUsername() Method**

```
public String getUsername()
```

Return the username.

**Returns**

*the username.*

---

## setArchiveDirectory(String) Method

```
public void setArchiveDirectory(String archiveDir)
```

Set the archive directory. Email messages and attachments are saved to this directory after the email is read and processed. The filenames of the archived files are prefixed with the timestamp of the time they were processed.

### Parameters

*archiveDir*

– the archive directory to set.

---

## setAttachmentAction(String) Method

```
public void setAttachmentAction(String attachmentAction)
```

Sets the attachment action. Attachments that are sent in the email messages can be either 'IGNORE'd or 'ARCHIVE'd. The default value is ARCHIVE.

### Parameters

*attachmentAction*

– the value to set.

---

## setChannel(String) Method

```
public void setChannel(String channel)
```

Set the name of the channel. Email messages are published to the channel specified by the channel name.

### Parameters

*channel*

– the name to channel to set.

---

## setComment(String) Method

```
public void setComment(String comment)
```

Set a description for the channel rule.

### Parameters

*comment*

– the description of this channel.

---

## setErrorDirectory(String) Method

```
public void setErrorDirectory(String errorDir)
```

Set the error directory. The body of the email message along with any attachments are copied to this directory if any errors are encountered when publishing the email message. The filenames of the archived files are prefixed with the timestamp of the time that the error occurred.

### Parameters

*errorDir*

– the error directory to set.

---

## setHostname(String) Method

```
public void setHostname(String hostName)
```

Set the hostname. This is the address of the email server to poll for messages.

### Parameters

*hostName*

– the hostname to set.

---

## setImapMoveFolder(String) Method

```
public void setImapMoveFolder(String imapMoveFolder)
```

Set the IMAP move folder name. Specify the name of the folder to move messages to after they are processed. This option is valid only if the protocol is set to IMAP.

### Parameters

*imapMoveFolder*

– the folder to move messages to.

---

## setPassword(String) Method

```
public void setPassword(String password)
```

Set the password to the email account being polled.

### Parameters

*password*

– the password to set.

---

## setPasswordAlias(String) Method

```
public void setPasswordAlias(String pwdAlias)
```

Set the password alias. Specify the alias to the password store in order to look up the password to the email account.

### Parameters

*pwdAlias*

– the alias to set.

---

## setPollingInterval(int) Method

```
public void setPollingInterval(int pollingInterval)
```

Set the polling interval. The polling interval specifies how often the email server is polled to check for new messages.

### Parameters

*pollingInterval*

– the polling interval in seconds.

---

## setPortNumber(int) Method

```
public void setPortNumber(int portNumber)
```

Set the port number on which the email server is accessible. The default value is –1 which allows the correct port number to be selected, depending on the protocol specified.

### Parameters

*portNumber*

– the port number to set.

---

## setPostReadAction(String) Method

```
public void setPostReadAction(String postReadAction)
```

Set the post-read action. Specify what to do with the email after it is read and published. The allowed values for a POP3 server are 'DELETE' and 'ARCHIVE'. An IMAP server allows 'DELETE', 'ARCHIVE' and 'MOVE'. The default value is 'DELETE'.

## Parameters

*postReadAction*

– the post read action to set.

---

## setProtocol(String) Method

```
public void setProtocol(String protocol)
```

Set the protocol to use to access the specified mailbox. The valid values are IMAP or POP3. Defaults to POP3.

## Parameters

*protocol*

– the protocol to set.

---

## setPublishAsUser(String) Method

```
public void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

## Parameters

*publishAsUser*

property to set

---

## setReadLimit(int) Method

```
public void setReadLimit(int readLimit)
```

Set the read limit. Specify the number of email messages to read in each polling of the email account. The default value is '0', which means no limit. All the messages in the Inbox are read in each polling interval.

## Parameters

*readLimit*

– the read limit to set.

---

## setSubjectFilter(String) Method

**DEPRECATED** Set the subject filter to filter email messages.

```
public void setSubjectFilter(String subjectFilter)
```

**Parameters***subjectFilter*

the subject filter to set.

---

**setUsername(String) Method**

```
public void setUsername(String username)
```

Set the username of the email account that needs to be polled for incoming messages.

**Parameters***username*

– the username of the account to poll.

---

**toString() Method**

```
public String toString()
```

Returns a string representation of this channel.

**Overrides**`Object.toString()`

# EmailEventGenConfigurationMBean Interface

*public interface* **EmailEventGenConfigurationMBean**

extends ConfigurationMBean

Manages channel rules for a Email Event Generator.

## Related Topics

EmailEventGeneratorsMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public
    EmailEventGenChannelConfiguration[] getChannels()
        Get all the channels for this event generator

    public String
        getEventGeneratorName()
        Get the name of the event generator

    public void
        setChannels(EmailEventGenChannelConfiguration[]
        channelsData)
        Set all the channels for this event generator
  
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## TYPE

```
public static final String TYPE
```



## Method Detail

### getChannels() Method

```
public EmailEventGenChannelConfiguration[] getChannels()
```

Get all the channels for this event generator

#### Returns

*the Channels for this event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### setChannels(EmailEventGenChannelConfiguration[]) Method

```
public void setChannels(EmailEventGenChannelConfiguration[] channelsData)  
    throws MBeanException
```

Set all the channels for this event generator

#### Parameters

*channelsData*  
Channels to add

#### Exceptions

*MBeanException*

# EmailEventGeneratorsMBean Interface

*public interface EmailEventGeneratorsMBean*

extends ConfigurationMBean

Provides a central access point for getting/creating Email Event Generator MBeans

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public void
        deleteEmailEventGenConfigurationMBean(EmailEventGenConfigurationMBean
        mbean)
        Delete a EmailEventGenConfigurationMBean

    public
    EmailEventGenConfigurationMBean[] getEmailEventGenConfigurationMBeans()
        Get a list of all known
        EmailEventGenConfigurationMBean's

    public
    EmailEventGenConfigurationMBean newEmailEventGenConfigurationMBean(String
    name)
        Create a new EmailEventGenConfigurationMBean
  
```

## Methods from interface weblogic.management.configuration.ConfigurationMBean

freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### **deleteEmailEventGenConfigurationMBean(EmailEventGenConfigurationMBean)** **Method**

```
public void deleteEmailEventGenConfigurationMBean(EmailEventGenConfigurationMBean mbean)
    throws MBeanException
```

Delete a EmailEventGenConfigurationMBean

#### **Parameters**

*mbean*  
MBean to delete

#### **Exceptions**

*MBeanException*

---

### **getEmailEventGenConfigurationMBeans() Method**

```
public EmailEventGenConfigurationMBean[] getEmailEventGenConfigurationMBeans()
    throws MBeanException
```

Get a list of all known EmailEventGenConfigurationMBean's

#### **Returns**

*Array of MBean's*

#### **Exceptions**

*MBeanException*

---

### **newEmailEventGenConfigurationMBean(String) Method**

```
public EmailEventGenConfigurationMBean newEmailEventGenConfigurationMBean(String name)
    throws MBeanException
```

Create a new EmailEventGenConfigurationMBean

#### **Parameters**

*name*  
of the new MBean

## Returns

*the created MBean*

## Exceptions

*MBeanException*

# FileEventGenChannelConfiguration Class

*public class FileEventGenChannelConfiguration*

extends Object  
implements Serializable

Represents a channel rule for a File Event Generator.

Rule Fields:

- directory – directory to scan
- scanSubdirs – scan subdirs or not
- filePattern – pattern of file to match

Other fields:

- channel – channel to send to if rules match
- comment – comment for this channel rule
- isXml – is the document XML or not (true, false)
- pollingInterval – how often to poll directory, in seconds
- sortByArrival – many files could have been written into the directory since the last polling sweep. If set to true then it means sort the files by arrival time “this maintains sequence (true, false)

## Related Topics

FileEventGenConfigurationMBean

---

## Hierarchy

Object  
FileEventGenChannelConfiguration

## All Implemented Interfaces

Serializable

---

## Field Summary

```
public
static _CHANNELTYPE_STRING
final String
String

public
static _CHANNELTYPE_XML
final String
String

public
static
```

```

final String _DEFAULT_CHANNELTYPE
        String

        public
        static _DEFAULT_FILE_MASK
final String
        String

        public
        static _DEFAULT_POLLING_INTERVAL
final int
        int

        public
        static _POSTREAD_ARCHIVE
final String
        String

        public
        static _POSTREAD_DELETE
final String
        String

```

## Constructor Summary

*FileEventGenChannelConfiguration()*

## Method Summary

```

public
String getArchiveDirectory()
        Get the archive–directory attribute

public
String getChannel()
        Get the name of the channel

public
String getChannelType()
        Get the is–xml attribute

public
String getComment()
        Get the comment attribute

public
String getEncoding()
        Get the encoding attribute

public
String getErrorDirectory()
        Get the error–directory attribute

public
String getFileDirectory()
        Get the fileDirectory Attribute

```

```

public getFilePattern()
    String          Get the file-pattern attribute

public
String getFtpHostName()
    Get the ftp-host-name attribute

public
String getFtpLocalDirectory()
    Get the ftp-local-directory attribute

public
String getFtpPassword()
    Get the ftp-user-password attribute

public
String getFtpPasswordAlias()
    Get the ftp-user-password-alias attribute

public
String getFtpUserName()
    Get the ftp-user-name attribute

public
boolean getPassByFileName()
    Get the pass-by-filename attribute

public
    int getPollingInterval()
        Get the polling-interval attribute

public
String getPostRead()
    Get the post-read attribute

public
String getPublishAsUser()
    Get the publishAsUser attribute

public
    int getReadLimit()
        Get the read-limit attribute

public
boolean getScanSubdirs()
    Get the scan-subdirectories attribute

public
boolean getSortByArrival()
    Get the sort-by-arrival attribute

public
    void setArchiveDirectory(String archiveDirectory)
        Set the archiveDirectory attribute

public
    void setChannel(String channel)
        Set the channel attribute

public
    void setChannelType(String channelType)
        Set the isXML attribute

```



```
public setComment(String comment)  
    void          Set the comment attribute  
public  
    void setEncoding(String encoding)  
        Set the encoding attribute  
public  
    void setErrorDirectory(String errorDirectory)  
        Set the errorDirectory attribute  
public  
    void setFileDirectory(String fileDirectory)  
        Set the fileDirectory attribute  
public  
    void setFilePattern(String filePattern)  
        Set the filePattern attribute  
public  
    void setFtpHostName(String ftpHostName)  
        Set the ftp-host-name attribute  
public  
    void setFtpLocalDirectory(String ftpLocalDirectory)  
        Set the ftp-local-directory attribute  
public  
    void setFtpPassword(String ftpPassword)  
public  
    void setFtpPasswordAlias(String ftpPasswordAlias)  
public  
    void setFtpUserName(String ftpUserName)  
        Set the ftp-user-name attribute  
public  
    void setPassByFileName(boolean passByFileName)  
        Set the passByFileName attribute  
public  
    void setPollingInterval(int pollingInterval)  
        Set the pollingInterval attribute  
public  
    void setPostRead(String postRead)  
        Set the postRead attribute  
public  
    void setPublishAsUser(String publishAsUser)  
        Set the publishAsUser attribute  
public  
    void setReadLimit(int readLimit)  
        Set the readLimit attribute  
public  
    void setScanSubdirs(boolean scanSubdirs)  
        Set the setScanSubdirs attribute  
public  
    void setSortByArrival(boolean sortByArrival)
```

Set the sortByArrival attribute

```
public
String toString()
```

Returns a string representation of this channel

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### **`_CHANNELTYPE_STRING`**

```
public static final String _CHANNELTYPE_STRING
```

---

### **`_CHANNELTYPE_XML`**

```
public static final String _CHANNELTYPE_XML
```

---

### **`_DEFAULT_CHANNELTYPE`**

```
public static final String _DEFAULT_CHANNELTYPE
```

---

### **`_DEFAULT_FILE_MASK`**

```
public static final String _DEFAULT_FILE_MASK
```

---

### **`_DEFAULT_POLLING_INTERVAL`**

```
public static final int _DEFAULT_POLLING_INTERVAL
```

---

### **`_POSTREAD_ARCHIVE`**

```
public static final String _POSTREAD_ARCHIVE
```

---

### **`_POSTREAD_DELETE`**

```
public static final String _POSTREAD_DELETE
```

## Constructor Detail

### FileEventGenChannelConfiguration

```
public FileEventGenChannelConfiguration()
```

## Method Detail

### getArchiveDirectory() Method

```
public String getArchiveDirectory()
```

Get the archive-directory attribute

#### Returns

*archive-directory attribute*

---

### getChannel() Method

```
public String getChannel()
```

Get the name of the channel

#### Returns

*name of channel*

---

### getChannelType() Method

```
public String getChannelType()
```

Get the is-xml attribute

#### Returns

*is-xml attribute*

---

### getComment() Method

```
public String getComment()
```

Get the comment attribute

**Returns**

*comment attribute*

---

**getEncoding() Method**

```
public String getEncoding()
```

Get the encoding attribute

**Returns**

*encoding attribute*

---

**getErrorDirectory() Method**

```
public String getErrorDirectory()
```

Get the error-directory attribute

**Returns**

*error-directory attribute*

---

**getFileDirectory() Method**

```
public String getFileDirectory()
```

Get the fileDirectory Attribute

**Returns**

*fileDirectory attribute*

---

**getFilePattern() Method**

```
public String getFilePattern()
```

Get the file-pattern attribute

**Returns**

*file-pattern attribute*

---

## getFtpHostName() Method

```
public String getFtpHostName()
```

Get the ftp-host-name attribute

### Returns

*ftp-host-name attribute*

---

## getFtpLocalDirectory() Method

```
public String getFtpLocalDirectory()
```

Get the ftp-local-directory attribute

### Returns

*ftp-local-directory attribute*

---

## getFtpPassword() Method

```
public String getFtpPassword()
```

Get the ftp-user-password attribute

### Returns

*ftp-user-password attribute*

---

## getFtpPasswordAlias() Method

```
public String getFtpPasswordAlias()
```

Get the ftp-user-password-alias attribute

### Returns

*ftp-user-password-alias attribute*

---

## getFtpUserName() Method

```
public String getFtpUserName()
```

Get the ftp-user-name attribute

**Returns**

*ftp-user-name attribute*

---

**getPassByFileName() Method**

```
public boolean getPassByFileName()
```

Get the pass-by-filename attribute

**Returns**

*pass-by-filename attribute*

---

**getPollingInterval() Method**

```
public int getPollingInterval()
```

Get the polling-interval attribute

**Returns**

*polling-interval attribute*

---

**getPostRead() Method**

```
public String getPostRead()
```

Get the post-read attribute

**Returns**

*post-read attribute*

---

**getPublishAsUser() Method**

```
public String getPublishAsUser()
```

Get the publishAsUser attribute

**Returns**

*publishAsUser attribute*

---

## getReadLimit() Method

```
public int getReadLimit()
```

Get the read-limit attribute

### Returns

*read-limit attribute*

---

## getScanSubdirs() Method

```
public boolean getScanSubdirs()
```

Get the scan-subdirectories attribute

### Returns

*scan-subdirectories attribute*

---

## getSortByArrival() Method

```
public boolean getSortByArrival()
```

Get the sort-by-arrival attribute

### Returns

*sort-by-arrival attribute*

---

## setArchiveDirectory(String) Method

```
public void setArchiveDirectory(String archiveDirectory)
```

Set the archiveDirectory attribute

### Parameters

*archiveDirectory*  
specified the directory to be used for archiving files.

---

## setChannel(String) Method

```
public void setChannel(String channel)
```

Set the channel attribute

**Parameters***channel*the channel for the event generator

---

**setChannelType(String) Method**

```
public void setChannelType(String channelType)
```

Set the isXML attribute

**Parameters***channelType*indicates whether the file type is XML or not.

---

**setComment(String) Method**

```
public void setComment(String comment)
```

Set the comment attribute

**Parameters***comment*is a user comment.

---

**setEncoding(String) Method**

```
public void setEncoding(String encoding)
```

Set the encoding attribute

**Parameters***encoding*character set encoding needed for String types.

---

**setErrorDirectory(String) Method**

```
public void setErrorDirectory(String errorDirectory)
```

Set the errorDirectory attribute



**Parameters***errorDirectory*

specified the directory to be used for files in error.

**setFileDirectory(String) Method**

```
public void setFileDirectory(String fileDirectory)
```

Set the fileDirectory attribute

**Parameters***fileDirectory*

the directory to be processed by the event generator

**setFilePattern(String) Method**

```
public void setFilePattern(String filePattern)
```

Set the filePattern attribute

**Parameters***filePattern*

the file pattern to be used to select files.

**setFtpHostName(String) Method**

```
public void setFtpHostName(String ftpHostName)
```

Set the ftp-host-name attribute

**Parameters***ftpHostName*

name of the ftp host

**setFtpLocalDirectory(String) Method**

```
public void setFtpLocalDirectory(String ftpLocalDirectory)
```

Set the ftp-local-directory attribute

**Parameters***ftpLocalDirectory*

local directory name to be used for ftp use.

**setFtpPassword(String) Method**

```
public void setFtpPassword(String ftpPassword)
```

**setFtpPasswordAlias(String) Method**

```
public void setFtpPasswordAlias(String ftpPasswordAlias)
```

**setFtpUserName(String) Method**

```
public void setFtpUserName(String ftpUserName)
```

Set the ftp-user-name attribute

**Parameters***ftpUserName*

name of the ftp user

**setPassByFileName(boolean) Method**

```
public void setPassByFileName(boolean passByFileName)
```

Set the passByFileName attribute

**Parameters***passByFileName*

indicates whether to publish the file name as opposed to the file content.

**setPollingInterval(int) Method**

```
public void setPollingInterval(int pollingInterval)
```

Set the pollingInterval attribute

**Parameters***pollingInterval*

indicates how often the directory to be scanned (in seconds).

## setPostRead(String) Method

```
public void setPostRead(String postRead)
```

Set the postRead attribute

### Parameters

*postRead*

specifies whether to delete or archive files after they are processed.

---

## setPublishAsUser(String) Method

```
public void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

### Parameters

*publishAsUser*

property to set

---

## setReadLimit(int) Method

```
public void setReadLimit(int readLimit)
```

Set the readLimit attribute

### Parameters

*readLimit*

indicates the number of files to be processed per scan.

---

## setScanSubdirs(boolean) Method

```
public void setScanSubdirs(boolean scanSubdirs)
```

Set the setScanSubdirs attribute

### Parameters

*scanSubdirs*

indicates whether to scan subdirectories or not.

---

## setSortByArrival(boolean) Method

```
public void setSortByArrival(boolean sortByArrival)
```

Set the sortByArrival attribute

### Parameters

*sortByArrival*

indicates whether to process files in the arrival time order.

---

## toString() Method

```
public String toString()
```

Returns a string representation of this channel

### Overrides

Object.toString()

# FileEventGenConfigurationMBean Interface

*public interface FileEventGenConfigurationMBean*

extends ConfigurationMBean

Manages channel rules for a File Event Generator.

## Related Topics

FileEventGeneratorsMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
    The magic number that is used to generate the serialVersionUID of the generated
    MBean Stub (xxxMBean_Stub).

public
static TYPE
final String
    String
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public
FileEventGenChannelConfiguration[] getChannels()
    Get all the channels for this event generator

public String
    getEventGeneratorName()
    Get the name of the event generator

public void
    setChannels(FileEventGenChannelConfiguration[]
    channelsData)
    Set all the channels for this event generator
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## TYPE

```
public static final String TYPE
```

## Method Detail

### getChannels() Method

```
public FileEventGenChannelConfiguration[] getChannels()
```

Get all the channels for this event generator

#### Returns

*the Channels for this event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### setChannels(FileEventGenChannelConfiguration[]) Method

```
public void setChannels(FileEventGenChannelConfiguration[] channelsData)  
    throws MBeanException
```

Set all the channels for this event generator

#### Parameters

*channelsData*  
Channels to add

#### Exceptions

*MBeanException*

# FileEventGeneratorsMBean Interface

*public interface FileEventGeneratorsMBean*

extends ConfigurationMBean

Provides a central access point for getting/creating File Event Generator MBeans

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

public static final long	<b>CACHING_STUB_SVUID</b>	The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean_Stub).
public static final String	<b>TYPE</b>	String

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

public void	<b>deleteFileEventGenConfigurationMBean</b> (FileEventGenConfigurationMBean mbean)	Delete a FileEventGenConfigurationMBean
public FileEventGenConfigurationMBean[]	<b>getFileEventGenConfigurationMBeans</b> ()	Get a list of all known FileEventGenConfigurationMBean's
public FileEventGenConfigurationMBean	<b>newFileEventGenConfigurationMBean</b> (String name)	Create a new FileEventGenConfigurationMBean

## Methods from interface weblogic.management.configuration.ConfigurationMBean



freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### deleteFileEventGenConfigurationMBean(FileEventGenConfigurationMBean) Method

```
public void deleteFileEventGenConfigurationMBean(FileEventGenConfigurationMBean mbean)
    throws MBeanException
```

Delete a FileEventGenConfigurationMBean

#### Parameters

*mbean*  
MBean to delete

#### Exceptions

*MBeanException*

---

### getFileEventGenConfigurationMBeans() Method

```
public FileEventGenConfigurationMBean[] getFileEventGenConfigurationMBeans()
    throws MBeanException
```

Get a list of all known FileEventGenConfigurationMBean's

#### Returns

*Array of MBean's*

#### Exceptions

*MBeanException*

---

### newFileEventGenConfigurationMBean(String) Method

```
public FileEventGenConfigurationMBean newFileEventGenConfigurationMBean(String name)
    throws MBeanException
```

Create a new FileEventGenConfigurationMBean

#### Parameters

*name*  
of the new MBean

## Returns

*the created MBean*

## Exceptions

*MBeanException*

# GenericEventGenChannelConfiguration Class

*public abstract class GenericEventGenChannelConfiguration*

extends Object  
implements Serializable

---

## **Hierarchy**

Object  
GenericEventGenChannelConfiguration

## **All Implemented Interfaces**

Serializable

---

## Constructor Summary

### ***GenericEventGenChannelConfiguration()***

Constructs a new GenericEventGenChannelConfiguration object.

## Method Summary

```
public
String getChannel()
    Return the channel name.

public
String getComment()
    Return the description.

public
String getPublishAsUser()
    Get the publishAsUser attribute

public
void setChannel(String channel)
    Set the name of the channel.

public
void setComment(String comment)
    Set a description for the channel rule.

public
void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

```
public  
String toString()
```

Returns a string representation of this channel.

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

### GenericEventGenChannelConfiguration

```
public GenericEventGenChannelConfiguration()
```

Constructs a new GenericEventGenChannelConfiguration object.

## Method Detail

### getChannel() Method

```
public String getChannel()
```

Return the channel name.

#### Returns

*the name of the channel.*

---

### getComment() Method

```
public String getComment()
```

Return the description.

---

### getPublishAsUser() Method

```
public String getPublishAsUser()
```

Get the publishAsUser attribute

**Returns**

*publishAsUser* attribute

---

**setChannel(String) Method**

```
public void setChannel(String channel)
```

Set the name of the channel. XEG messages are published to the channel specified by the channel name.

**Parameters**

*channel*

– the name to channel to set.

---

**setComment(String) Method**

```
public void setComment(String comment)
```

Set a description for the channel rule.

**Parameters**

*comment*

– the description of this channel.

---

**setPublishAsUser(String) Method**

```
public void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

**Parameters**

*publishAsUser*

property to set

---

**toString() Method**

```
public String toString()
```

Returns a string representation of this channel.

**Overrides**

Object.toString()



# GenericEventGenConfigurationMBean Interface

*public interface GenericEventGenConfigurationMBean*

extends ConfigurationMBean

Manages channel rules for a Generic Event Generator.

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public
    GenericEventGenChannelConfiguration[] getChannels()
        Get all the channels for this event generator

    public String
        getEventGenConfigTypeName()

    public String
        getEventGeneratorName()
        Get the name of the event generator

    public void
        setChannels(GenericEventGenChannelConfiguration[] channelsData)
        Set all the channels for this event generator
  
```

## Methods from interface weblogic.management.configuration.ConfigurationMBean



freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

### TYPE

```
public static final String TYPE
```

## Method Detail

### getChannels() Method

```
public GenericEventGenChannelConfiguration[] getChannels()
```

Get all the channels for this event generator

#### Returns

*the Channels for this event generator*

---

### getEventGenConfigTypeName() Method

```
public String getEventGenConfigTypeName()
```

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### setChannels(GenericEventGenChannelConfiguration[]) Method

```
public void setChannels(GenericEventGenChannelConfiguration[] channelsData)  
    throws MBeanException
```

Set all the channels for this event generator

#### Parameters

*channelsData*  
Channels to add

#### Exceptions

*MBeanException*

# GenericEventGeneratorsMBean Interface

*public interface GenericEventGeneratorsMBean*

extends ConfigurationMBean

Provides a central access point for getting/creating Event Generator MBeans

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public void
        deleteEventGenConfigurationMBean(GenericEventG
        mbean)
        Delete a GenericEventGenConfigurationMBean

    public
    GenericEventGenConfigurationMBean[] getEventGenConfigurationMBeans()
        Get a list of all known
        GenericEventGenConfigurationMBean's

    public
    GenericEventGenConfigurationMBean newEventGenConfigurationMBean(String
        name)
        Create a new
        GenericEventGenConfigurationMBean
  
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## TYPE

```
public static final String TYPE
```

## Method Detail

### deleteEventGenConfigurationMBean(GenericEventGenConfigurationMBean) Method

```
public void deleteEventGenConfigurationMBean(GenericEventGenConfigurationMBean mbean)  
    throws MBeanException
```

Delete a GenericEventGenConfigurationMBean

#### Parameters

*mbean*  
MBean to delete

#### Exceptions

*MBeanException*

---

### getEventGenConfigurationMBeans() Method

```
public GenericEventGenConfigurationMBean[] getEventGenConfigurationMBeans()  
    throws MBeanException
```

Get a list of all known GenericEventGenConfigurationMBean's

#### Returns

*Array of MBean's*

#### Exceptions

*MBeanException*

---

### newEventGenConfigurationMBean(String) Method

```
public GenericEventGenConfigurationMBean newEventGenConfigurationMBean(String name)  
    throws MBeanException
```

Create a new GenericEventGenConfigurationMBean

#### Parameters

*name*  
of the new MBean

## Returns

*the created MBean*

## Exceptions

*MBeanException*

# JMSEventGenChannelConfiguration Class

*public class JMSEventGenChannelConfiguration*

extends Object  
implements Serializable

Represents a channel rule for a JMS Event Generator.

Rule Fields:

- matchJmsProperty, matchJmsPropertyValue
  - ◆ If the jms property indicated by matchJmsPropertyName matches matchJmsPropertyValue, thatâ€™s a match.
  - ◆ If matchJmsPropertyName is configured but matchJmsPropertyValue isnâ€™t, the presence of the property will trigger the match.
  - ◆ If matchJmsPropertyName is absent, thatâ€™s always a match (like the default in a case statement)

Other fields:

- channel – channel to send to if rules match
- comment – comment for this channel rule
- packageType – package type to send (xml, RawData, String)

## Related Topics

JMSEventGenConfigurationMBean

---

## Hierarchy

Object  
JMSEventGenChannelConfiguration

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*JMSEventGenChannelConfiguration()*

## Method Summary

```

public
String getChannel()
    Get the name of the channel

public
String getComment()
    Get the comment attribute

public
String getMatchJmsPropertyName()
    Get the matchJmsPropertyName Attribute

public
String getMatchJmsPropertyValue()
    Get the matchJmsPropertyValues attribute

public
String getPublishAsUser()
    Get the publishAsUser attribute

public
    void setChannel(String channel)
        Set the channel attribute

public
    void setComment(String comment)
        Set the comment attribute

public
    void setMatchJmsPropertyName(String matchJmsPropertyName)
        Set the matchJmsProperty attribute

public
    void setMatchJmsPropertyValue(String matchJmsPropertyValue)
        Set the matchJmsValue attribute

public
    void setPublishAsUser(String publishAsUser)
        Set the publishAsUser attribute

public
String toString()
    Returns a string representation of this channel

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail



## JMSEventGenChannelConfiguration

```
public JMSEventGenChannelConfiguration()
```

### Method Detail

#### getChannel() Method

```
public String getChannel()
```

Get the name of the channel

##### Returns

*name of channel*

---

#### getComment() Method

```
public String getComment()
```

Get the comment attribute

##### Returns

*comment attribute*

---

#### getMatchJmsPropertyName() Method

```
public String getMatchJmsPropertyName()
```

Get the matchJmsPropertyName Attribute

##### Returns

*matchJmsPropertyName attribute*

---

#### getMatchJmsPropertyValue() Method

```
public String getMatchJmsPropertyValue()
```

Get the matchJmsPropertyValues attribute

**Returns**

*matchJmsPropertyValue* attribute

---

**getPublishAsUser() Method**

```
public String getPublishAsUser()
```

Get the publishAsUser attribute

**Returns**

*publishAsUser* attribute

---

**setChannel(String) Method**

```
public void setChannel(String channel)
```

Set the channel attribute

**Parameters**

*channel*

the channel for the event generator

---

**setComment(String) Method**

```
public void setComment(String comment)
```

Set the comment attribute

**Parameters**

*comment*

is a user comment.

---

**setMatchJmsPropertyName(String) Method**

```
public void setMatchJmsPropertyName(String matchJmsPropertyName)
```

Set the matchJmsProperty attribute

**Parameters**

*matchJmsPropertyName*

property to set

## setMatchJmsPropertyValue(String) Method

```
public void setMatchJmsPropertyValue(String matchJmsPropertyValue)
```

Set the matchJmsValue attribute

### Parameters

*matchJmsPropertyValue*  
property to set

---

## setPublishAsUser(String) Method

```
public void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

### Parameters

*publishAsUser*  
property to set

---

## toString() Method

```
public String toString()
```

Returns a string representation of this channel

### Overrides

Object.toString()

# JMSEventGenConfigurationMBean Interface

*public interface JMSEventGenConfigurationMBean*

extends ConfigurationMBean

Manages channel rules for a JMS Event Generator.

## Related Topics

JMSEventGeneratorsMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
    The magic number that is used to generate the serialVersionUID of the generated
    MBean Stub (xxxMBean_Stub).

public
static TYPE
final String
    String
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public
JMSEventGenChannelConfiguration[] getChannels()
    Get all the channels for this event generator

public String
getEventGeneratorName()
    Get the name of the event generator

public String[]
getForeignJmsDestinationNames()
    Get a list of JNDI names of JMS foreign destination
    name in the system
```

## Java Class Reference

```
public String[] getJmsQueueNames()
    Get a list of JNDI names of JMS queues name in the
    system

public String[]
    getJmsTopicNames()
    Get a list of JNDI names of JMS topics name in the
    system

public void
    setChannels(JMSEventGenChannelConfiguration[]
    channelsData)
    Set all the channels for this event generator
```

### Methods from interface **weblogic.management.configuration.ConfigurationMBean**

freezeCurrentValue, getAttributeStringValue, getComments, getNotes,  
getSetFields, getXml, getXmlConverter, isDefaultedMBean,  
isPersistenceEnabled, preDeregister, registerConfigMBean,  
restoreDefaultValue, setComments, setDefaultedMBean, setNotes,  
setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface **javax.management.DynamicMBean**

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,  
setAttributes

### Methods from interface **javax.management.MBeanRegistration**

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface **javax.management.NotificationBroadcaster**

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface **weblogic.management.WebLogicMBean**

getMBeanInfo, getName, getObjectNames, getParent, getType,  
isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.

– change the signature of an existing "boolean getter", i.e, "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getChannels() Method

```
public JMSEventGenChannelConfiguration[] getChannels()
```

Get all the channels for this event generator

#### Returns

*the Channels for this event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### getForeignJmsDestinationNames() Method

```
public String[] getForeignJmsDestinationNames()
```

Get a list of JNDI names of JMS foreign destination name in the system

#### Returns

*list of JMS destinations*

---

## getJmsQueueNames() Method

```
public String[] getJmsQueueNames()
```

Get a list of JNDI names of JMS queues name in the system

### Returns

*list of JMS queues*

---

## getJmsTopicNames() Method

```
public String[] getJmsTopicNames()
```

Get a list of JNDI names of JMS topics name in the system

### Returns

*list of JMS topics*

---

## setChannels(JMSEventGenChannelConfiguration[]) Method

```
public void setChannels(JMSEventGenChannelConfiguration[] channelsData)  
    throws MBeanException
```

Set all the channels for this event generator

### Parameters

*channelsData*  
Channels to add

### Exceptions

*MBeanException*

# JMSEventGeneratorsMBean Interface

*public interface JMSEventGeneratorsMBean*

extends ConfigurationMBean

Provides a central access point for getting/creating JMS Event Generator MBeans

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public void
        deleteJMSEventGenConfigurationMBean(JMSEventGenCon
        mbean)
        Delete a JMSEventGenConfigurationMBean

    public String[]
        getForeignJmsDestinationNames()
        Get a list of JNDI names of JMS foreign destination
        name in the system

    public
    JMSEventGenConfigurationMBean[] getJMSEventGenConfigurationMBeans()
        Get a list of all known
        JMSEventGenConfigurationMBean's

    public String[]
        getJmsQueueNames()
  
```



## Java Class Reference

Get a list of JNDI names of JMS queues name in the system

```
public String[]
```

```
    getJmsTopicNames()
```

Get a list of JNDI names of JMS topics name in the system

```
    public
```

```
JMSEventGenConfigurationMBean newJMSEventGenConfigurationMBean(String name)  
    Create a new JMSEventGenConfigurationMBean
```

### Methods from interface **weblogic.management.configuration.ConfigurationMBean**

```
freezeCurrentValue, getAttributeStringValue, getComments, getNotes,  
getSetFields, getXml, getXmlConverter, isDefaultedMBean,  
isPersistenceEnabled, preDeregister, registerConfigMBean,  
restoreDefaultValue, setComments, setDefaultedMBean, setNotes,  
setPersistenceEnabled, touch, unregisterConfigMBean
```

### Methods from interface **javax.management.DynamicMBean**

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,  
setAttributes
```

### Methods from interface **javax.management.MBeanRegistration**

```
postDeregister, postRegister, preDeregister, preRegister
```

### Methods from interface **javax.management.NotificationBroadcaster**

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

### Methods from interface **weblogic.management.WebLogicMBean**

```
getMBeanInfo, getName, getObjectNames, getParent, getType,  
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### **CACHING\_STUB\_SVUID**

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

### TYPE

```
public static final String TYPE
```

## Method Detail

### deleteJMSEventGenConfigurationMBean(JMSEventGenConfigurationMBean) Method

```
public void deleteJMSEventGenConfigurationMBean(JMSEventGenConfigurationMBean mbean)
    throws MBeanException
```

Delete a JMSEventGenConfigurationMBean

#### Parameters

*mbean*  
MBean to delete

#### Exceptions

*MBeanException*

---

### getForeignJmsDestinationNames() Method

```
public String[] getForeignJmsDestinationNames()
```

Get a list of JNDI names of JMS foreign destination name in the system

#### Returns

*list of JMS destinations*

---

### getJMSEventGenConfigurationMBeans() Method

```
public JMSEventGenConfigurationMBean[] getJMSEventGenConfigurationMBeans()
    throws MBeanException
```

Get a list of all known JMSEventGenConfigurationMBean's

**Returns**

*Array of MBean's*

**Exceptions**

*MBeanException*

---

**getJmsQueueNames() Method**

```
public String[] getJmsQueueNames()
```

Get a list of JNDI names of JMS queues name in the system

**Returns**

*list of JMS queues*

---

**getJmsTopicNames() Method**

```
public String[] getJmsTopicNames()
```

Get a list of JNDI names of JMS topics name in the system

**Returns**

*list of JMS topics*

---

**newJMSEventGenConfigurationMBean(String) Method**

```
public JMSEventGenConfigurationMBean newJMSEventGenConfigurationMBean(String name)  
    throws MBeanException
```

Create a new JMSEventGenConfigurationMBean

**Parameters**

*name*  
of the new MBean

**Returns**

*the created MBean*

**Exceptions**

*MBeanException*



# MessageBrokerConfigurationMBean Interface

*public interface MessageBrokerConfigurationMBean*

extends ConfigurationMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

public  
static **CACHING\_STUB\_SVUID**  
final long

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

public  
void **deployChannelPublishPolicy**(String channel, Set roles)  
Registers a new publish policy on a channel.

public  
void **deployChannelSubscriptionPolicy**(String channel, Set roles)  
Registers a new subscription policy on a channel.

public  
String **getChannelDispatchAsPrincipal**(String channel)  
Returns the dispatch-as principal for the channel.

public  
Set **getChannelPublishPolicy**(String channel)  
Returns the channel's publish policy.

public  
Set **getChannelSubscribePolicy**(String channel)  
Returns the channel's subscribe policy.

public  
void **setChannelDispatchAsPrincipal**(String channel, String principal)  
Sets the channel's dispatch-as principal.

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### **CACHING\_STUB\_SVUID**

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### deployChannelPublishPolicy(String, Set) Method

```
public void deployChannelPublishPolicy(String channel,
                                       Set roles)
    throws SecurityException
```

Registers a new publish policy on a channel.

#### Exceptions

*SecurityException*

---

### deployChannelSubscriptionPolicy(String, Set) Method

```
public void deployChannelSubscriptionPolicy(String channel,
                                           Set roles)
    throws SecurityException
```

Registers a new subscription policy on a channel.

#### Exceptions

*SecurityException*

---

### getChannelDispatchAsPrincipal(String) Method

```
public String getChannelDispatchAsPrincipal(String channel)
```

Returns the dispatch-as principal for the channel. If no dispatch-as has been explicitly specified, null is returned.

#### Returns

*the dispatch-as principal.*

---

### getChannelPublishPolicy(String) Method

```
public Set getChannelPublishPolicy(String channel)
```

Returns the channel's publish policy.

#### Returns

*an array of those global roles allowed to publish to the channel.*

---

## **getChannelSubscribePolicy(String) Method**

```
public Set getChannelSubscribePolicy(String channel)
```

Returns the channel's subscribe policy.

### **Returns**

*an array of those global roles allowed to subscribe to the channel.*

---

## **setChannelDispatchAsPrincipal(String, String) Method**

```
public void setChannelDispatchAsPrincipal(String channel,  
                                           String principal)
```

Sets the channel's dispatch-as principal.



# NodeMethodInfo Class

*public class NodeMethodInfo*

extends Object  
implements Serializable

Entry into map from a process's method name to information about its method. This includes the associated control's class name and whether the method has a return value. This is used by the process graph, as it is created at runtime and lacks certain information.

---

## Hierarchy

Object  
NodeMethodInfo

## All Implemented Interfaces

Serializable

---

## Field Summary

```
public
static final int DTF_CONTROL
int
```

## Constructor Summary

*NodeMethodInfo(int controlType, boolean hasReturn)*

Create a NodeMethodInfo record

*NodeMethodInfo(Class ctlClass, boolean hasReturn)*

Create a NodeMethodInfo record

## Method Summary

```
public
static String findCtlIconId(Class ctlClass)
```

```

    public getIconId()
        String          Get this NodeMethnodInfo's iconID
    public
    static
    ByteBuffer getNodeMethodInfo(Class iconInterface)
    public
    boolean hasReturn()
        Flags whether the method has a retrun value, which means node is a
        controlSendReturn node.
    public
    boolean isStaticIconFile()

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### DTF\_CONTROL

```
public static final int DTF_CONTROL
```

## Constructor Detail

### NodeMethodInfo

```
public NodeMethodInfo(int controlType,
                     boolean hasReturn)
```

Create a NodeMethodInfo record

---

### NodeMethodInfo

```
public NodeMethodInfo(Class ctlClass,
                     boolean hasReturn)
```

Create a NodeMethodInfo record

## Method Detail

### findCtlIconId(Class) Method

```
public static String findCtlIconId(Class ctlClass)
```

#### Parameters

*ctlClass*

a controls Java class

#### Returns

*known control Java interface name. Used as key to find icon*

---

### getIconId() Method

```
public String getIconId()
```

Get this NodeMethodInfo's iconID

#### Returns

*string that identifies the iconId. This is implemented as the interface name of the control having the icon.*

---

### getNodeMethodInfo(Class) Method

```
public static ByteBuffer getNodeMethodInfo(Class iconInterface)
```

#### Parameters

*iconInterface*

Java interface name for a known control

#### Returns

*a ByteBuffer which is the icon file which represents controls implementing iconInterface*

---

### hasReturn() Method

```
public boolean hasReturn()
```

Flags whether the method has a return value, which means node is a controlSendReturn node.

---

## **isStaticIconFile() Method**

```
public boolean isStaticIconFile()
```

# PasswordStoreConfigurationMBean Interface

*public interface PasswordStoreConfigurationMBean*

extends ConfigurationMBean

Manages the WLI password store

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

public static final long	<b>CACHING_STUB_SVUID</b>	The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean_Stub).
public static final String	<b>TYPE</b>	String

Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

public void	<b>deletePassword</b> (String alias)	delete a password
public String[]	<b>getAliasList</b> ()	Get all the aliases in the store
public String	<b>getPassword</b> (String alias)	Get a password
public void	<b>setPassword</b> (String alias, String password)	Set a password
public void	<b>updatePassword</b> (String alias, String oldPassword, String newPassword)	

update a password

#### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

#### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

#### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

#### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

#### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### deletePassword(String) Method

```
public void deletePassword(String alias)  
    throws NotFoundException, MBeanException
```

delete a password

#### Parameters

*alias*  
Alias of password

#### Exceptions

*NotFoundException*  
*MBeanException*

---

### getAliasList() Method

```
public String[] getAliasList()  
    throws MBeanException
```

Get all the aliases in the store

#### Returns

*array of aliases*

#### Exceptions

*MBeanException*

---

### getPassword(String) Method

```
public String getPassword(String alias)  
    throws NotFoundException, MBeanException
```

Get a password

**Parameters***alias*

Alias of password

**Returns***password***Exceptions***NotFoundException**MBeanException*

---

**setPassword(String, String) Method**

```
public void setPassword(String alias,  
                        String password)  
    throws AlreadyExistsException, MBeanException
```

Set a password

**Parameters***alias*

Alias of password

*password*

Password to set

**Exceptions***AlreadyExistsException**MBeanException*

---

**updatePassword(String, String, String) Method**

```
public void updatePassword(String alias,  
                           String oldPassword,  
                           String newPassword)  
    throws NotFoundException, MBeanException
```

update a password

**Parameters***alias*

Alias of password



*oldPassword*

the old password

*newPassword*

the new password

## **Exceptions**

*NotFoundException*

*MBeanException*

# ProcessConfigurationMBean Interface

*public interface* **ProcessConfigurationMBean**

extends ConfigurationMBean

MBean for managing properties of a particular **type** of process. This mbean exposes certain configurable properties of a process type and allows them to be dynamically changed. There is one instance of ProcessConfigurationMBean for each service type. Use ProcessRuntimeMBean to manage instances of a process.

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration, NotificationBroadcaster, WebLogicMBean

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public long
    getActivationTime()
        Return the activation time in seconds since 1/1/1970 GMT.

public String
    getAppName()
        Returns the application name where this process is deployed.

public
ArchivingPolicy getArchivingPolicy()
    Returns the archiving policy for this process.

public String
    getCallbackClassName()
        Returns the class name for the client callback if there is any.

public Set
    getControlCallbackPolicy(String controlId)
```

## Java Class Reference

Returns the callback authorization policy for a given control.

```
public long
```

***getDeactivationTime()***  
Returns the deactivation time in seconds since 1/1/1970 GMT.

```
public String
```

***getDeploymentId()***  
Return id which is different for each deployment of this process.

```
public String
```

***getDescription()***  
Returns the description for this process type.

```
public String
```

***getDisplayName()***  
Returns the display name.

```
public String[]
```

***getExternalCallbackControls()***  
Returns the names of all controls in this process which are external callback targets.

```
public String
```

***getFlowTag()***  
Returns the flow definition in String format.

```
public String
```

***getInheritedPolicy(String methodName)***  
Returns the effective authorization policy expression for the given method when no policy has been explicitly assigned to the method.

```
public Map
```

***getMethodControlInfo()***  
Return info about a method's associated control.

```
public Set
```

***getMethodPolicy(String methodName)***  
Returns the method authorization policy.

```
public
```

```
ProcessNodeInfo[] getNodes()
```

Returns an array of all the nodes in the process definition.

```
public
```

```
ProcessNodeInfo[] getNodes(int type)
```

Returns an array of nodes of a certain type.

```
public Set
```

***getProcessPolicy()***  
Returns the process-level authorization policy.

```
public
```

```
ProcessVariable getProcessVariable(String varName)
```

Returns the type information for the given process variable.

```
public
```

```
ProcessVariable[] getProcessVariables()
```

Returns type information about process variables (excluding controls, callbacks, and other special variables).

```
public String
```

```

        getPublicURI()
            Returns the public URI of this process type.
public String
        getRelativeEJBPath()
            Returns the relative EJB path where the ejb for this process resides.
public String
        getRunAs()
            Returns the run-as attribute for this process.
public String
        getServiceURI()
            Returns the service of the process type represented by this MBean.
    public URL
        getServiceURL()
            Returns the service URL for this process.
public long
        getSLA()
            Returns the SLA in milliseconds.
public double
        getSLAWarningThreshold()
            Returns the SLA warning threshold as fraction of the actual SLA in
            milliseconds.
    public
TrackingLevel getTrackingLevel()
            Returns the tracking level for this process.
public String
        getTypeName()
            Returns the type name (name of the class implementing this process).
public String
        getWebappName()
            Returns the name of the webapp where this process is deployed.
public boolean
        hasDeclarativeAuthorization()
            Returns true if: 1) there is a roles-allowed annotation at the class level, 2)
            there is a callback-roles-allowed annotation at the class level or, 3) there is a
            roles-allowed annotation on at least one method in this process.
public boolean
        isExecutionPolicyModifiable()
            Returns false if there is a common:security annotation on the jpd file,
            otherwise returns true.
public boolean
        isRunnable()
            Returns whether the process is runnable.
public boolean
        isSinglePrincipalRequired()
            Returns true iff this process requires that all incoming client requests come
            from the same principal.
public boolean

```

```

isStateless()
    Returns whether the process is stateless.

public boolean
isVersioned()
    Returns true if this process is part of a version group.

public void
setActivationAttributes(boolean runnable, long activationTime, long
deactivationTime)
    Set all activation attributes for the process.

public void
setActivationTime(long time)
    Set the activation time in seconds since 1/1/1970 GMT.

public void
setArchivingPolicy(ArchivingPolicy archivingPolicy)
    Sets the archiving policy for this process type.

public void
setControlCallbackPolicy(String controlId, Set roles)
    Sets the callback authorization policy for a given control.

public void
setDeactivationTime(long time)
    Set the deactivation time in seconds since 1/1/1970 GMT.

public void
setDisplayName(String displayName)
    Sets the display name.

public void
setMethodPolicy(String methodName, Set roles)
    Sets the method authorization policy.

public void
setProcessPolicy(Set roles)
    Sets the process-level authorization policy.

public void
setRunAs(String runAs)
    Sets the run-as attribute for this process.

public void
setRunnable(boolean runnable)
    Enable or disable process.

public void
setSinglePrincipalRequired(boolean flag)
    Sets the single-principal flag for this process.

public void
setSLA(long sla)
    Sets the SLA.

public void
setSLAWarningThreshold(double ratio)
    Sets the SLA warning threshold.

public void
setTrackingLevel(TrackingLevel trackingLevel)

```

Sets the tracking level for this process type.

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### getActivationTime() Method

```
public long getActivationTime()
```

Return the activation time in seconds since 1/1/1970 GMT.

---

### getAppName() Method

```
public String getAppName()  
    throws ProcessNotFoundException
```

Returns the application name where this process is deployed.

#### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

### getArchivingPolicy() Method

```
public ArchivingPolicy getArchivingPolicy()  
    throws MBeanException
```

Returns the archiving policy for this process.

#### Exceptions

*MBeanException*

---

### getCallbackClassName() Method

```
public String getCallbackClassName()  
    throws ProcessNotFoundException
```

Returns the class name for the client callback if there is any.

#### Returns

*callback class name, null if there is no callback*

#### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## getControlCallbackPolicy(String) Method

```
public Set getControlCallbackPolicy(String controlId)  
    throws MBeanException, ProcessNotFoundException
```

Returns the callback authorization policy for a given control.

### Parameters

*controlId*

the id of an external callback target control on this process.

### Returns

*the set of roles authorized at the control callback points.*

### Exceptions

*MBeanException*

if there is any other error getting the policy

*ProcessNotFoundException*

if the process is not found

---

## getDeactivationTime() Method

```
public long getDeactivationTime()
```

Returns the deactivation time in seconds since 1/1/1970 GMT.

### Returns

*the deactivation time in seconds since 1/1/1970 GMT.*

---

## getDeploymentId() Method

```
public String getDeploymentId()  
    throws ProcessNotFoundException
```

Return id which is different for each deployment of this process. Internal use only.

### Exceptions

*ProcessNotFoundException*

if the process is not found

---



## getDescription() Method

```
public String getDescription()  
    throws ProcessNotFoundException
```

Returns the description for this process type. This is the same as the description text entered for the top level node in the workshop.

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## getDisplayName() Method

```
public String getDisplayName()  
    throws MBeanException
```

Returns the display name. This is mainly used for pretty printing in the Console.

### Exceptions

*MBeanException*

---

## getExternalCallbackControls() Method

```
public String[] getExternalCallbackControls()  
    throws ProcessNotFoundException
```

Returns the names of all controls in this process which are external callback targets. These controls can have authorization policies assigned to their callbacks. Returns an empty array if there are no external callback target controls in this process.

### Returns

*an array of control names.*

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## getFlowTag() Method

```
public String getFlowTag()  
    throws ProcessNotFoundException, ProcessNotFoundException
```

Returns the flow definition in String format.

**Exceptions**

*ProcessNotFoundException*  
if the process is not found  
*ProcessNotFoundException*  
if the process is not found

---

**getInheritedPolicy(String) Method**

```
public String getInheritedPolicy(String methodName)
    throws MBeanException, ProcessNotFoundException, NoSuchMethodException
```

Returns the effective authorization policy expression for the given method when no policy has been explicitly assigned to the method.

**Returns**

*a policy expression*

**Exceptions**

*MBeanException*  
*ProcessNotFoundException*  
if the process is not found  
*NoSuchMethodException*  
if there is no method by the given name.

---

**getMethodControlInfo() Method**

```
public Map getMethodControlInfo()
    throws ProcessNotFoundException
```

Return info about a method's associated control. Currently only relevant for controlSend nodes. Used to get correct icon for process graph. Internal use only.

**Exceptions**

*ProcessNotFoundException*  
if the process is not found

---

**getMethodPolicy(String) Method**

```
public Set getMethodPolicy(String methodName)
    throws MBeanException, ProcessNotFoundException, NoSuchMethodException
```

Returns the method authorization policy. The policy is a set of roles. If a method policy is specified, a user is allowed to invoke the method if he belongs to at least one of the roles. This method will return an empty array if no method-level policy has been set.

**Returns**

*an array of role names (the allowed roles), or an empty array if no method-level policy has been set.*

**Exceptions**

*MBeanException*

*ProcessNotFoundException*

if the process is not found

*NoSuchMethodException*

if there is no method by the given name.

---

**getNodes() Method**

```
public ProcessNodeInfo[] getNodes()
    throws ProcessNotFoundException
```

Returns an array of all the nodes in the process definition.

**Exceptions**

*ProcessNotFoundException*

if the process is not found

---

**getNodes(int) Method**

```
public ProcessNodeInfo[] getNodes(int type)
    throws ProcessNotFoundException
```

Returns an array of nodes of a certain type.

**Parameters**

*type*

the type of the nodes to be returned. Types are defined in

**Returns**

*an array of nodes of the given type. If there are no nodes of that type an empty array is returned.*

**Exceptions**

*ProcessNotFoundException*

if the process is not found

---

## getProcessPolicy() Method

```
public Set getProcessPolicy()
    throws MBeanException, ProcessNotFoundException
```

Returns the process-level authorization policy. The policy is a set of roles. A user is allowed to invoke an operation on the process if he belongs to at least one of those roles. This policy applies to all operations on the process, but individual methods can expand the policy by allowing additional roles, effectively relaxing restrictions (see `ProcessConfigurationMBean.setMethodPolicy(String, Set)`).

### Returns

*an array of role names (the allowed roles), or an empty array if no process-level policy has been set.*

### Exceptions

*MBeanException*  
*ProcessNotFoundException*  
 if the process is not found

---

## getProcessVariable(String) Method

```
public ProcessVariable getProcessVariable(String varName)
    throws MBeanException, ProcessNotFoundException
```

Returns the type information for the given process variable.

### Parameters

*varName*  
 name of the variable

### Exceptions

*MBeanException*  
*ProcessNotFoundException*  
 if the process is not found

---

## getProcessVariables() Method

```
public ProcessVariable[] getProcessVariables()
    throws MBeanException, ProcessNotFoundException
```

Returns type information about process variables (excluding controls, callbacks, and other special variables).

### Exceptions

*MBeanException*  
*ProcessNotFoundException*

if the process is not found

---

## getPublicURI() Method

```
public String getPublicURI()
```

Returns the public URI of this process type. If the process is a part of a version group then the group URI of the version group is returned. Otherwise the service URI is returned as the public URI

---

## getRelativeEJBPath() Method

```
public String getRelativeEJBPath()  
    throws ProcessNotFoundException
```

Returns the relative EJB path where the ejb for this process resides.

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## getRunAs() Method

```
public String getRunAs()  
    throws MBeanException, ProcessNotFoundException
```

Returns the run-as attribute for this process. This is either: <start-user>, null or a principal name. A null return value means invocations run under the caller's identity.

### Returns

*the run-as attribute.*

### Exceptions

*MBeanException*  
*ProcessNotFoundException*  
if the process is not found

---

## getServiceURI() Method

```
public String getServiceURI()
```

Returns the service of the process type represented by this MBean.

---

## getServiceURL() Method

```
public URL getServiceURL()  
    throws ProcessNotFoundException
```

Returns the service URL for this process.

### Exceptions

*ProcessNotFoundException*

---

## getSLA() Method

```
public long getSLA()  
    throws MBeanException
```

Returns the SLA in milliseconds. A return value that is less than or equal to 0 indicates that SLA is not defined for this process type

### Exceptions

*MBeanException*

---

## getSLAWarningThreshold() Method

```
public double getSLAWarningThreshold()  
    throws MBeanException
```

Returns the SLA warning threshold as fraction of the actual SLA in milliseconds. A return value that is less than or equal to 0.0 indicates that SLA warning is not defined for this process type

### Exceptions

*MBeanException*

---

## getTrackingLevel() Method

```
public TrackingLevel getTrackingLevel()  
    throws MBeanException
```

Returns the tracking level for this process.

### Exceptions

*MBeanException*

---

## getTypeName() Method

```
public String getTypeName()  
    throws ProcessNotFoundException
```

Returns the type name (name of the class implementing this process).

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## getWebappName() Method

```
public String getWebappName()  
    throws ProcessNotFoundException
```

Returns the name of the webapp where this process is deployed.

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## hasDeclarativeAuthorization() Method

```
public boolean hasDeclarativeAuthorization()  
    throws ProcessNotFoundException
```

Returns true if: 1) there is a roles-allowed annotation at the class level, 2) there is a callback-roles-allowed annotation at the class level or, 3) there is a roles-allowed annotation on at least one method in this process. Otherwise returns false. If the jpd file has declarative security then authorization policies cannot be set dynamically.

### Returns

*true/false*

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## isExecutionPolicyModifiable() Method

```
public boolean isExecutionPolicyModifiable()  
    throws ProcessNotFoundException
```

Returns false if there is a common:security annotation on the jpd file, otherwise returns true.

### Returns

*true/false*

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---

## isRunnable() Method

```
public boolean isRunnable()
```

Returns whether the process is runnable.

---

## isSinglePrincipalRequired() Method

```
public boolean isSinglePrincipalRequired()  
    throws MBeanException, ProcessNotFoundException
```

Returns true iff this process requires that all incoming client requests come from the same principal.

### Returns

*true/flase*

### Exceptions

*MBeanException*  
*ProcessNotFoundException*  
if the process is not found

---

## isStateless() Method

```
public boolean isStateless()  
    throws ProcessNotFoundException
```

Returns whether the process is stateless.

### Exceptions

*ProcessNotFoundException*  
if the process is not found

---



## isVersioned() Method

```
public boolean isVersioned()
```

Returns `true` if this process is part of a version group.

---

## setActivationAttributes(boolean, long, long) Method

```
public void setActivationAttributes(boolean runnable,  
                                   long activationTime,  
                                   long deactivationTime)
```

Set all activation attributes for the process.

### Parameters

*runnable*

is this process runnable?

*activationTime*

time from which the process will be active. (in seconds since 1/1/1970 GMT)

*deactivationTime*

time at which the process cease to be active. (in seconds since 1/1/1970 GMT)

---

## setActivationTime(long) Method

```
public void setActivationTime(long time)
```

Set the activation time in seconds since 1/1/1970 GMT.

### Parameters

*time*

time in seconds since 1/1/1970 GMT.

---

## setArchivingPolicy(ArchivingPolicy) Method

```
public void setArchivingPolicy(ArchivingPolicy archivingPolicy)  
    throws MBeanException, InvalidAttributeValueException
```

Sets the archiving policy for this process type.

### Parameters

*archivingPolicy*

the new archiving policy.

**Exceptions***MBeanException**InvalidAttributeValueException*

---

**setControlCallbackPolicy(String, Set) Method**

```
public void setControlCallbackPolicy(String controlId,
                                     Set roles)
    throws MBeanException, ProcessNotFoundException
```

Sets the callback authorization policy for a given control. If there are any roles–allowed annotations on the jpd file, authorization policies cannot be dynamically changed.

**Parameters***controlId*

the id of an external callback target control on this process.

*roles*

the set of roles authorized at the control callback points.

**Exceptions***MBeanException*

if policies are read–only, or if there is no external callback target control by the given controlId.

*ProcessNotFoundException*

if the process is not found

---

**setDeactivationTime(long) Method**

```
public void setDeactivationTime(long time)
```

Set the deactivation time in seconds since 1/1/1970 GMT.

**Parameters***time*

deactivation time in seconds since 1/1/1970

---

**setDisplayDisplayName(String) Method**

```
public void setDisplayName(String displayName)
    throws MBeanException
```

Sets the display name. This can be any short descriptive name.

## Exceptions

*MBeanException*

---

### setMethodPolicy(String, Set) Method

```
public void setMethodPolicy(String methodName,
                           Set roles)
    throws MBeanException, ProcessNotFoundException, NoSuchMethodException
```

Sets the method authorization policy. The policy is a set of roles. If there is a method policy, a user is allowed to invoke an operation on the process if he belongs to at least one of the roles. Note that if there is a process-level policy, all the roles in the process policy are added to the roles specified in this call to determine the actual method policy. Calling this method with an empty array clears the process policy (subject, of course, to the process policy). If there are any roles-allowed annotations on the jpd file, authorization policies cannot be dynamically set.

## Exceptions

*MBeanException*

if policies are read-only

*ProcessNotFoundException*

if the process is not found

*NoSuchMethodException*

or if there is no method by the given name.

---

### setProcessPolicy(Set) Method

```
public void setProcessPolicy(Set roles)
    throws MBeanException, ProcessNotFoundException
```

Sets the process-level authorization policy. The policy is a set of roles. A user is allowed to invoke an operation on the process if he belongs to at least one of those roles. This policy applies to all operations on the process, but individual methods can expand the policy by allowing additional roles, effectively relaxing restrictions (see `ProcessConfigurationMBean.setMethodPolicy(String, Set)`). Calling this method with an empty array clears the process policy. If there are any roles-allowed annotations on the jpd file, authorization policies cannot be dynamically set. NOTE: when changing the process policy, any roles in the new process policy which were not present in the overwritten process policy are automatically added to all method policies. On the other hand, dropping roles from the process policy does not cause any side-effect on the method policies.

## Parameters

*roles*

an array of role names.

**Exceptions***MBeanException*

if policies are read-only.

*ProcessNotFoundException*

if the process is not found

**setRunAs(String) Method**

```
public void setRunAs(String runAs)
    throws MBeanException, ProcessNotFoundException
```

Sets the run-as attribute for this process. A null argument represents the "callers-identity" execution policy (this is the default policy). Note that if there is a common:security annotation on the jpd file, then the run-as attribute is read-only and any attempt to call this method will throw an exception.

**Parameters***runAs*

either "&lt;start-user&gt;", a WLS username or null.

**Exceptions***MBeanException*

if the argument is invalid or if there is a common:security annotation in the jpd file.

*ProcessNotFoundException*

if the process is not found

**setRunnable(boolean) Method**

```
public void setRunnable(boolean runnable)
```

Enable or disable process.

**setSinglePrincipalRequired(boolean) Method**

```
public void setSinglePrincipalRequired(boolean flag)
    throws MBeanException, ProcessNotFoundException
```

Sets the single-principal flag for this process. Note that this can only be set if it was not already set via an annotation on the jpd file.

**Parameters***flag*

the new single-principal flag

## Exceptions

*MBeanException*

if there is already a single-principal annotation on the jpd.

*ProcessNotFoundException*

if the process is not found

---

## setSLA(long) Method

```
public void setSLA(long sla)
    throws MBeanException
```

Sets the SLA.

### Parameters

*sla*

SLA in milliseconds. A value that is less than or equal to 0 unsets the SLA

## Exceptions

*MBeanException*

---

## setSLAWarningThreshold(double) Method

```
public void setSLAWarningThreshold(double ratio)
    throws MBeanException
```

Sets the SLA warning threshold.

### Parameters

*ratio*

SLA warning threshold as a fraction of actual SLA. A value that is less than or equal to 0 unsets the SLA warning.

## Exceptions

*MBeanException*

---

## setTrackingLevel(TrackingLevel) Method

```
public void setTrackingLevel(TrackingLevel trackingLevel)
    throws MBeanException, InvalidAttributeValueException, ProcessNotFoundException
```

Sets the tracking level for this process type.

## Parameters

*trackingLevel*  
the new tracking level.

## Exceptions

*MBeanException*  
*InvalidAttributeValueException*  
*ProcessNotFoundException*  
if the process is not found

# ProcessControlMBean Interface

*public interface* **ProcessControlMBean**

extends ConfigurationMBean

MBean for managing properties of a Process Control

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public void
    addPropertiesForSelector(String selectorValue,
        String targetURI, String userName, String
        password)
    Adds a new selector value and associated initial
    values for various properties.

public String
    getContextPath()
    Returns the context-path of the webapp that
    contains the Process Control this MBean represents

public String
    getControlClass()
    Returns the fully-qualified class name of the
    Process Control this MBean represents

public
ProcessControlPropertiesDocument getProperties(String selectorValue)
    Returns the dynamic properties for the control
    represented by this MBean and a specific selector
```

## Java Class Reference

```
value
public String[]
    getSelectorValues()
        Returns all the selector values defined for this
        Process Control
public void
    removeSelector(String selectorValue)
        removes the given selector value and all properties
        associated with it.
public void
    setProperties(String selectorValue,
        ProcessControlPropertiesDocument props)
        Sets the dynamic properties for the control
        represented by this MBean, for the specific selector
        value
```

### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`,  
`getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`,  
`isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`,  
`restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`,  
`setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`,  
`setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectName`, `getParent`, `getType`,  
`isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this



interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### addPropertiesForSelector(String, String, String, String) Method

```
public void addPropertiesForSelector(String selectorValue,
                                    String targetURI,
                                    String userName,
                                    String password)
    throws MBeanException
```

Adds a new selector value and associated initial values for various properties. Throws MBeanException if the selector value already exists

#### Parameters

*selectorValue*  
new selector value

#### Exceptions

*MBeanException*

---

### getContextPath() Method

```
public String getContextPath()
```

Returns the context-path of the webapp that contains the Process Control this MBean represents

---

### getControlClass() Method

```
public String getControlClass()
```

Returns the fully-qualified class name of the Process Control this MBean represents

---

## getProperties(String) Method

```
public ProcessControlPropertiesDocument getProperties(String selectorValue)
    throws MBeanException
```

Returns the dynamic properties for the control represented by this MBean and a specific selector value

### Returns

*the dynamic properties or null if there is no entry matching the selector value*

### Exceptions

*MBeanException*

---

## getSelectorValues() Method

```
public String[] getSelectorValues()
```

Returns all the selector values defined for this Process Control

---

## removeSelector(String) Method

```
public void removeSelector(String selectorValue)
    throws MBeanException
```

removes the given selector value and all properties associated with it. Throws MBeanException if the selector does not exist

### Parameters

*selectorValue*  
– selector to be removed

### Exceptions

*MBeanException*

---

## setProperties(String, ProcessControlPropertiesDocument) Method

```
public void setProperties(String selectorValue,
    ProcessControlPropertiesDocument props)
    throws MBeanException
```

Sets the dynamic properties for the control represented by this MBean, for the specific selector value

## Exceptions

*MBeanException*

# ProcessNodeInfo Interface

*public interface* **ProcessNodeInfo**

extends `Serializable`

Represents a node in the process definition

---

## *All Superinterfaces*

`Serializable`

---

## Method Summary

```
public
    int[] getChildIds()
        Returns an array of ids of all the child nodes.

public
    int getId()
        Returns the id of the process node

public
    String getMethod()
        Returns the name of the java method that is invoked as a part of executing this node if
        any

public
    String getName()
        Returns the name of the process node

public
    int getParentId()
        Returns the id of the parent node.

public
    int getType()
        Returns the type of the process node as defined in
```

## Method Detail

### **getChildIds() Method**

```
public int[] getChildIds()
```

Returns an array of ids of all the child nodes.

## Returns

*an array of ids of all the child nodes. Returns an empty array if the node does not have any children.*

---

## getId() Method

```
public int getId()
```

Returns the id of the process node

## Returns

*the id of the process node*

---

## getMethod() Method

```
public String getMethod()
```

Returns the name of the java method that is invoked as a part of executing this node if any

## Returns

*the name of the java method*

---

## getName() Method

```
public String getName()
```

Returns the name of the process node

## Returns

*the name of the process node*

---

## getParentId() Method

```
public int getParentId()
```

Returns the id of the parent node.

## Returns

*the id of the parent node. Returns -1 if the node does not have parent*

---

## **getType() Method**

```
public int getType()
```

Returns the type of the process node as defined in

### **Returns**

*the type of the process node*

# ProcessVariable Class

*public class ProcessVariable*

extends Object  
implements Serializable

Represents a process variable with its name and type (class)

## Related Topics

ProcessConfigurationMBean.getProcessVariable(String)  
ProcessConfigurationMBean.getProcessVariables()

---

## Hierarchy

Object  
ProcessVariable

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*ProcessVariable(String name, String type)*

## Method Summary

public  
String **getName()**  
Returns the name of the process variable

public  
String **getType()**  
Returns the type of the process variable.

public  
String **toString()**

Methods from class `java.lang.Object`

`clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait`

## Constructor Detail

### ProcessVariable

```
public ProcessVariable(String name,  
                       String type)
```

## Method Detail

### getName() Method

```
public String getName()
```

Returns the name of the process variable

---

### getType() Method

```
public String getType()
```

Returns the type of the process variable. This method deliberately returns a `String` rather than a `Class` object, since the user of this API (e.g., the client) may not have the `Class` in its classpath.

---

### toString() Method

```
public String toString()
```

#### *Overrides*

`Object.toString()`



# SampleConfigurationMBean Interface

*public interface SampleConfigurationMBean*

extends ConfigurationMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

Fields from interface `weblogic.management.configuration.ConfigurationMBean`

CACHING\_STUB\_SVUID

## Method Summary

```
public  
String getFoo()  
public  
void setFoo(String f)
```

Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`,  
`getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`,  
`isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`,  
`restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`,  
`setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`,  
`setAttributes`

Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

**Methods from interface `weblogic.management.WebLogicMBean`**

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`,  
`isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Method Detail

### **`getFoo()` Method**

```
public String getFoo()
```

---

### **`setFoo(String)` Method**

```
public void setFoo(String f)
```

# ServiceBrokerControlMBean Interface

*public interface ServiceBrokerControlMBean*

extends ConfigurationMBean

MBean for managing properties of a Service Broker Control

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public void
    addPropertiesForSelector(String
        selectorValue,
        ServiceBrokerControlPropertiesDocument
        properties)
    Adds a new selector value and associated initial
    values for various properties.

public String
    getContextPath()
    Returns the context-path of the webapp that
    contains the Service Broker Control this MBean
    represents

public String
    getControlClass()
    Returns the fully-qualified class name of the
    Service Broker Control this MBean represents

public
ServiceBrokerControlPropertiesDocument getProperties(String selectorValue)
```

Returns the dynamic properties for the control represented by this MBean and a specific selector value

```
public String[]
    getSelectorValues()
        Returns all the selector values defined for this
        Configuration MBean
```

public boolean

```
    isCallback()
        Returns true if this represents a callback
        within a JPD file, rather than a standalone
        ServiceBrokerControl
```

public void

```
    removeSelector(String selectorValue)
        removes the given selector value and all
        properties associated with it.
```

public void

```
    setProperties(String selectorValue,
        ServiceBrokerControlPropertiesDocument
        props)
        Sets the dynamic properties for the control
        represented by this MBean, for the specific
        selector value
```

#### Methods from interface **weblogic.management.configuration.ConfigurationMBean**

freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

#### Methods from interface **javax.management.DynamicMBean**

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

#### Methods from interface **javax.management.MBeanRegistration**

postDeregister, postRegister, preDeregister, preRegister

#### Methods from interface **javax.management.NotificationBroadcaster**

addNotificationListener, getNotificationInfo, removeNotificationListener

#### Methods from interface **weblogic.management.WebLogicMBean**

getMBeanInfo, getName, getObjectName, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### addPropertiesForSelector(String, ServiceBrokerControlPropertiesDocument) Method

```
public void addPropertiesForSelector(String selectorValue,
                                   ServiceBrokerControlPropertiesDocument properties)
    throws MBeanException
```

Adds a new selector value and associated initial values for various properties. Throws MBeanException if the selector value already exists

#### Parameters

*selectorValue*  
new selector value

*properties*  
the dynamic properties associated with this selector value

#### Exceptions

*MBeanException*

---

### getContextPath() Method

```
public String getContextPath()
```

Returns the context–path of the webapp that contains the Service Broker Control this MBean represents

---

## getControlClass() Method

```
public String getControlClass()
```

Returns the fully–qualified class name of the Service Broker Control this MBean represents

---

## getProperties(String) Method

```
public ServiceBrokerControlPropertiesDocument getProperties(String selectorValue)
```

Returns the dynamic properties for the control represented by this MBean and a specific selector value

### Returns

*the dynamic properties or null if there is no entry matching the selector value*

---

## getSelectorValues() Method

```
public String[] getSelectorValues()
```

Returns all the selector values defined for this Configuration MBean

---

## isCallback() Method

```
public boolean isCallback()
```

Returns `true` if this represents a callback within a JPD file, rather than a standalone ServiceBrokerControl

---

## removeSelector(String) Method

```
public void removeSelector(String selectorValue)  
    throws MBeanException
```

removes the given selector value and all properties associated with it. Throws MBeanException if the selector does not exist

### Parameters

*selectorValue*

– selector to be removed

## Exceptions

*MBeanException*

---

## **setProperties(String, ServiceBrokerControlPropertiesDocument) Method**

```
public void setProperties(String selectorValue,  
                        ServiceBrokerControlPropertiesDocument props)
```

Sets the dynamic properties for the control represented by this MBean, for the specific selector value

# TimerEventGenChannelConfiguration Class

*public class TimerEventGenChannelConfiguration*

extends Object

implements Serializable

Represents a channel rule for a Timer Event Generator.

Rule Fields:

- effective – initial time to fire
- expiry – end time to fire
- frequency – how often to fire after effective

Other fields:

- channel – channel to send to if rules match
- comment – comment for this channel rule
- calendar-name – name of the business calendar used
- message-content – message to publish

## Related Topics

TimerEventGenConfigurationMBean

---

## Hierarchy

Object

TimerEventGenChannelConfiguration

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*TimerEventGenChannelConfiguration()*



## Method Summary

```

public
String getCalendarName()
    Returns the business calendar name of the channel rule

public
String getChannel()
    Returns the Message Broker channel name of the channel rule

public
String getComment()
    Returns the comment of the channel rule

public
String getEffective()
    Returns the initial time of the channel rule

public
    long getEffectiveTimestamp()
        Returns the initial time of the channel rule in milliseconds since January 1,
        1970, 00:00:00 GMT

public
String getExpiry()
    Returns the expiry time of the channel rule

public
    long getExpiryTimestamp()
        Returns the expiry time of the channel rule in milliseconds since January 1,
        1970, 00:00:00 GMT

public
String getFrequency()
    Returns the frequency of the channel rule

public
String getMessageContent()
    Returns the message content to publish of the channel rule

public
String getPublishAsUser()
    Get the publishAsUser attribute

public
    void setCalendarName(String calendarName)
        Set the business calendar name of the channel rule

public
    void setChannel(String channel)
        Set the Message Broker channel name of this channel rule

public
    void setComment(String comment)
        Set comment of the channel rule

public
    void setEffective(String effective)
        Set the initial time of the channel rule

```

```

public setExpiry(String expiry)
    void          Set the expiry time of the channel rule
public
    void setFrequency(String frequency)
        Set the frequency of the channel rule
public
    void setMessageContent(String messageContent)
        Set the message content to publish of the channel rule
public
    void setPublishAsUser(String publishAsUser)
        Set the publishAsUser attribute
public
String toString()
    Returns a string representation of this channel

```

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Constructor Detail

### TimerEventGenChannelConfiguration

```
public TimerEventGenChannelConfiguration()
```

## Method Detail

### getCalendarName() Method

```
public String getCalendarName()
```

Returns the business calendar name of the channel rule

#### Returns

*the business calendar name*

---

### getChannel() Method

```
public String getChannel()
```

Returns the Message Broker channel name of the channel rule

**Returns**

*the Message Broker channel name*

---

**getComment() Method**

```
public String getComment()
```

Returns the comment of the channel rule

**Returns**

*the comment*

---

**getEffective() Method**

```
public String getEffective()
```

Returns the initial time of the channel rule

**Returns**

*initial time*

---

**getEffectiveTimestamp() Method**

```
public long getEffectiveTimestamp()
```

Returns the initial time of the channel rule in milliseconds since January 1, 1970, 00:00:00 GMT

**Returns**

*initial time in milliseconds since January 1, 1970, 00:00:00 GMT*

---

**getExpiry() Method**

```
public String getExpiry()
```

Returns the expiry time of the channel rule

**Returns**

*expiry time*

---

## getExpiryTimestamp() Method

```
public long getExpiryTimestamp()
```

Returns the expiry time of the channel rule in milliseconds since January 1, 1970, 00:00:00 GMT

### Returns

*expiry time in milliseconds since January 1, 1970, 00:00:00 GMT*

---

## getFrequency() Method

```
public String getFrequency()
```

Returns the frequency of the channel rule

### Returns

*the frequency*

---

## getMessageContent() Method

```
public String getMessageContent()
```

Returns the message content to publish of the channel rule

### Returns

*the message to publish*

---

## getPublishAsUser() Method

```
public String getPublishAsUser()
```

Get the publishAsUser attribute

### Returns

*publishAsUser attribute*

---

## setCalendarName(String) Method

```
public void setCalendarName(String calendarName)
```

Set the business calendar name of the channel rule

**Parameters***calendarName*business calendar name

---

**setChannel(String) Method**

```
public void setChannel(String channel)
```

Set the Message Broker channel name of this channel rule

**Parameters***channel*name of the Message Broker channel

---

**setComment(String) Method**

```
public void setComment(String comment)
```

Set comment of the channel rule

**Parameters***comment*comment of the channel rule

---

**setEffective(String) Method**

```
public void setEffective(String effective)
```

Set the initial time of the channel rule

**Parameters***effective*the initial time

---

**setExpiry(String) Method**

```
public void setExpiry(String expiry)
```

Set the expiry time of the channel rule

**Parameters**

*expiry*  
the expiry time

---

**setFrequency(String) Method**

```
public void setFrequency(String frequency)
```

Set the frequency of the channel rule

**Parameters**

*frequency*  
how often to trigger the timer event

---

**setMessageContent(String) Method**

```
public void setMessageContent(String messageContent)
```

Set the message content to publish of the channel rule

**Parameters**

*messageContent*  
message to publish

---

**setPublishAsUser(String) Method**

```
public void setPublishAsUser(String publishAsUser)
```

Set the publishAsUser attribute

**Parameters**

*publishAsUser*  
property to set

---

**toString() Method**

```
public String toString()
```

Returns a string representation of this channel

**Overrides**

Object.toString()



# TimerEventGenConfigurationMBean Interface

*public interface* **TimerEventGenConfigurationMBean**

extends ConfigurationMBean

Manages channel rules for a Timer Event Generator.

## Related Topics

TimerEventGeneratorsMBean

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public
    TimerEventGenChannelConfiguration[] getChannels()
        Get all the channels for this event generator

    public String
        getEventGeneratorName()
        Get the name of the event generator

    public void
        setChannels(TimerEventGenChannelConfiguration[]
        channelsData)
        Set all the channels for this event generator
  
```



### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

`freezeCurrentValue`, `getAttributeStringValue`, `getComments`, `getNotes`, `getSetFields`, `getXml`, `getXmlConverter`, `isDefaultedMBean`, `isPersistenceEnabled`, `preDeregister`, `registerConfigMBean`, `restoreDefaultValue`, `setComments`, `setDefaultedMBean`, `setNotes`, `setPersistenceEnabled`, `touch`, `unRegisterConfigMBean`

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## TYPE

```
public static final String TYPE
```

## Method Detail

### getChannels() Method

```
public TimerEventGenChannelConfiguration[] getChannels()
```

Get all the channels for this event generator

#### Returns

*the Channels for this event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### setChannels(TimerEventGenChannelConfiguration[]) Method

```
public void setChannels(TimerEventGenChannelConfiguration[] channelsData)  
    throws MBeanException
```

Set all the channels for this event generator

#### Parameters

*channelsData*  
Channels to add

#### Exceptions

*MBeanException*

# TimerEventGeneratorsMBean Interface

*public interface* **TimerEventGeneratorsMBean**

extends ConfigurationMBean

Provides a central access point for getting/creating Timer Event Generator MBeans

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public void
        deleteTimerEventGenConfigurationMBean(TimerEvent
        mbean)
        Delete a TimerEventGenConfigurationMBean

    public
    TimerEventGenConfigurationMBean[] getTimerEventGenConfigurationMBeans()
        Get a list of all known
        TimerEventGenConfigurationMBean's

    public
    TimerEventGenConfigurationMBean newTimerEventGenConfigurationMBean(String
        name)
        Create a new TimerEventGenConfigurationMBean
  
```

## Methods from interface weblogic.management.configuration.ConfigurationMBean

freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### deleteTimerEventGenConfigurationMBean(TimerEventGenConfigurationMBean) Method

```
public void deleteTimerEventGenConfigurationMBean(TimerEventGenConfigurationMBean mbean)
    throws MBeanException
```

Delete a TimerEventGenConfigurationMBean

#### Parameters

*mbean*  
MBean to delete

#### Exceptions

*MBeanException*

---

### getTimerEventGenConfigurationMBeans() Method

```
public TimerEventGenConfigurationMBean[] getTimerEventGenConfigurationMBeans()
    throws MBeanException
```

Get a list of all known TimerEventGenConfigurationMBean's

#### Returns

*Array of MBean's*

#### Exceptions

*MBeanException*

---

### newTimerEventGenConfigurationMBean(String) Method

```
public TimerEventGenConfigurationMBean newTimerEventGenConfigurationMBean(String name)
    throws MBeanException
```

Create a new TimerEventGenConfigurationMBean

#### Parameters

*name*  
of the new MBean

## Returns

*the created MBean*

## Exceptions

*MBeanException*

# TrackingLevel Class

*public class TrackingLevel*

extends Object  
implements Serializable

Typesafe enumerated class for describing tracking levels for processes. Tracking level of a process indicates what data related to the execution of an instance will be logged.

---

## Hierarchy

```

Object
  TrackingLevel
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Field Summary

```

    public
    static DEFAULT
final TrackingLevel
    TrackingLevel indicating that the system wide default TrackingLevel should
    be used

    public
    static FULL
final TrackingLevel
    TrackingLevel for tracking event information and messages

    public
    static MINIMUM
final TrackingLevel
    TrackingLevel for tracking global events only, such as process start,end,
    suspend, resume, etc

    public
    static NONE
final TrackingLevel
    TrackingLevel for logging event information only, no messages are logged.

    public
    static NONE
final TrackingLevel
    TrackingLevel for not tracking any events

    public
    static VALUES
final List
    List of all TrackingLevel enumerated types.
  
```

## Method Summary

```

    public int
        intValue()
            Returns an integer representation for this TrackingLevel

    public
    String toString()
            Returns string representation of this TrackingLevel

    public
    static toTrackingLevel(int val)
TrackingLevel    Returns the Trackinglevel object whose integer representation is the given
                  integer value.

    public
    static toTrackingLevel(String s)
TrackingLevel    Returns the Trackinglevel object whose String representation is the given String
                  value.

```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `wait`, `wait`, `wait`

## Field Detail

### DEFAULT

```
public static final TrackingLevel DEFAULT
```

TrackingLevel indicating that the system wide default TrackingLevel should be used

---

### FULL

```
public static final TrackingLevel FULL
```

TrackingLevel for tracking event information and messages

---

### MINIMUM

```
public static final TrackingLevel MINIMUM
```

TrackingLevel for tracking global events only, such as process start,end, suspend, resume, etc

---



## NODE

```
public static final TrackingLevel NODE
```

TrackingLevel for logging event information only, no messages are logged.

---

## NONE

```
public static final TrackingLevel NONE
```

TrackingLevel for not tracking any events

---

## VALUES

```
public static final List VALUES
```

List of all TrackingLevel enumerated types.

## Method Detail

### intValue() Method

```
public int intValue()
```

Returns an integer representation for this TrackingLevel

### Related Topics

```
TrackingLevel.toTrackingLevel(int)
```

---

### toString() Method

```
public String toString()
```

Returns string representation of this TrackingLevel

### *Overrides*

Object.toString()

---

### toTrackingLevel(int) Method

```
public static TrackingLevel toTrackingLevel(int val)
```

Returns the Trackinglevel object whose integer representation is the given integer value.

---

### **toTrackingLevel(String) Method**

```
public static TrackingLevel toTrackingLevel(String s)
```

Returns the Trackinglevel object whose String representation is the given String value.

# UserInfoConfigurationMBean Interface

*public interface UserInfoConfigurationMBean*

extends ConfigurationMBean

Configuration MBean interface for Users in WLI.

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public
void deleteWLIUserInfo(String userid)
    Removes this user information from the WLI database

public
String getUserCalendarId(String userid)
    Retrieve the calendar id for this user

public
String getUserEmail(String userid)
    Retrieve the email address for this user.

public
void sync()
    Removes all users from the WLI database that do not exist in WLS

public
void unsetUserCalendar(String userid)
    Unsets the calendar for this user.

public
void updateUserCalendar(String userid, String calendarId)
```

Sets the calendar for this user.

```
public
    void updateUserEmail(String userid, String email)
        Sets the email address for this user.
```

#### Methods from interface `weblogic.management.configuration.ConfigurationMBean`

```
freezeCurrentValue, getAttributeStringValue, getComments, getNotes,
getSetFields, getXml, getXmlConverter, isDefaultedMBean,
isPersistenceEnabled, preDeregister, registerConfigMBean,
restoreDefaultValue, setComments, setDefaultedMBean, setNotes,
setPersistenceEnabled, touch, unregisterConfigMBean
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectName, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### deleteWLIUserInfo(String) Method

```
public void deleteWLIUserInfo(String userid)
    throws MBeanException
```

Removes this user information from the WLI database

#### Exceptions

*MBeanException*

---

### getUserCalendarId(String) Method

```
public String getUserCalendarId(String userid)
    throws MBeanException
```

Retrieve the calendar id for this user

#### Returns

*The calendar id for the user or null if there is no calendar.*

#### Exceptions

*MBeanException*

---

### getUserEmail(String) Method

```
public String getUserEmail(String userid)
    throws MBeanException
```

Retrieve the email address for this user.

#### Returns

*The email address of the user or null if there is no email address is found.*

#### Exceptions

*MBeanException*

---

### sync() Method

```
public void sync()
    throws MBeanException
```

Removes all users from the WLI database that do not exist in WLS

### Exceptions

*MBeanException*

---

### unsetUserCalendar(String) Method

```
public void unsetUserCalendar(String userid)
    throws MBeanException
```

Unsets the calendar for this user.

### Exceptions

*MBeanException*

---

### updateUserCalendar(String, String) Method

```
public void updateUserCalendar(String userid,
                               String calendarId)
    throws MBeanException
```

Sets the calendar for this user. A user record is created if it does not exist in the WLI database. Exception is thrown if the calendar does not exist or the user is not a valid WLS user.

### Exceptions

*MBeanException*

---

### updateUserEmail(String, String) Method

```
public void updateUserEmail(String userid,
                             String email)
    throws MBeanException
```

Sets the email address for this user. A user record is created if it does not exist in the WLI database. Exception is thrown if the user is not a valid WLS user.

### Exceptions

*MBeanException*

## VersioningConfigurationMBean Interface

*public interface VersioningConfigurationMBean*

extends ConfigurationMBean

Configuration MBean interface for Processes' Versioning.

Declare versioning information lookups and configuration settings at run-time.

Versioned processes are defined by:

- A Component group URI : public URI that is used as an alias to invoke the current active version. A Component group URI does not correspond to any physical process file.
- A set of component URIs: all the alternate versions that can be used for the group URI. Each component URI corresponds to an actual physical Process file.
- A default component URI: Component URI of the version used as reference for all versions of a group. The default component defines the minimum set of operations and callbacks a version must implement in order to be a valid alternate version for a group.

There is one instance of VersioningConfigurationMBean per versioned group. That is, each versioning MBean is identified by its service's group URI that can be accessed with :

`getGroup()`

You can get the current active component for a this group by calling:

`String activeURI = getActiveComponent()`.

For instance, the resulting activeURI could be `"/webappName/relativePath/process_v1.jspd"` which would be one of the component URI registered for this group.

Additionally, this MBean can access and set the attributes of each component as well as the group's attributes such as:

- the interface URI of the group
- the set of component URIs that belongs to the group
- the active flag of a component
- the activation date of a component
- the deactivation date of a component

Once you modified the configuration, you may want to check that it is still valid. Run the `validate()` method to ensure that the modifications did not break the versioning integrity.

Note that this MBean does not allow to add or remove a version component.

---

### *All Superinterfaces*

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

### Fields from interface `weblogic.management.configuration.ConfigurationMBean`

`CACHING_STUB_SVUID`

## Method Summary

```
public
long getActivationTime(String componentURI)
    Get the activation time for a particular component

public
String getActiveComponent()
    Get the current runtime active component for this MBean's group The current active
    component for a group is selected as follows:
    • its active flag is "true"
    • it has the most recent activation date among all the active components in its group
    • it complies to the interface defined by the default component of the group

public
String[] getComponentArray()
    Get the array of component URIs that belongs to the component group

public
long getDeactivationTime(String componentURI)
    Get the deactivation time for a particular component

public
String getDefaultComponent()
    Get the String URI of the default component (or interface) for a particular group

public
String getGroup()
    Get the component group URI for this MBean

public
boolean isEnabled(String componentURI)
    Is the component enabled?

public
boolean isVersioningEnabled()
    Is Version Manager enabled and initialized?

public
void set(String componentURI, boolean active, long activationDate)
    Set both the activation date and the active flag of a particular component.

public
void set(String componentURI, boolean active, long activationDate, long deactivationDate)
```



Set both the activation date and the active flag of a particular component.

```
public
    void setActivationTime(String componentURI, long activationTime)
        Set the activation time of a particular component , keep the same activation date.

public
    void setDeactivationTime(String componentURI, long deactivationTime)
        Set the activation time of a particular component , keep the same activation date.

public
    void setEnabled(String componentURI, boolean active)
        Enable or disable a particular component active attribute

public
String[] validate()
    Validate the current Version configuration, keep the same active flag A configuration is
    valid if there is only one currently active version or if there are multiple active versions
    and each of them has distinct activation dates.
```

### Methods from interface **weblogic.management.configuration.ConfigurationMBean**

```
freezeCurrentValue, getAttributeStringValue, getComments, getNotes,
getSetFields, getXml, getXmlConverter, isDefaultedMBean,
isPersistenceEnabled, preDeregister, registerConfigMBean,
restoreDefaultValue, setComments, setDefaultedMBean, setNotes,
setPersistenceEnabled, touch, unregisterConfigMBean
```

### Methods from interface **javax.management.DynamicMBean**

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

### Methods from interface **javax.management.MBeanRegistration**

```
postDeregister, postRegister, preDeregister, preRegister
```

### Methods from interface **javax.management.NotificationBroadcaster**

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

### Methods from interface **weblogic.management.WebLogicMBean**

```
getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### **CACHING\_STUB\_SVUID**

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub

(xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e, "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### getActivationTime(String) Method

```
public long getActivationTime(String componentURI)
    throws VersionException
```

Get the activation time for a particular component

#### Parameters

*componentURI*  
String URI of component

#### Returns

*activation time*

#### Exceptions

*VersionException*

---

### getActiveComponent() Method

```
public String getActiveComponent()
    throws VersionException
```

Get the current runtime active component for this MBean's group The current active component for a group is selected as follows:

- its active flag is "true"
- it has the most recent activation date among all the active components in its group
- it complies to the interface defined by the default component of the group

**Returns**

*String URI of current active component or null if none is currently active*

**Exceptions**

*VersionException*

---

**getComponentArray() Method**

```
public String[] getComponentArray()  
    throws VersionException
```

Get the array of component URIs that belongs to the component group

**Returns**

*String array of component URIs*

**Exceptions**

*VersionException*

---

**getDeactivationTime(String) Method**

```
public long getDeactivationTime(String componentURI)  
    throws VersionException
```

Get the deactivation time for a particular component

**Parameters**

*componentURI*  
String URI of component

**Returns**

*deactivation time*

**Exceptions**

*VersionException*

---

**getDefaultComponent() Method**

```
public String getDefaultComponent()  
    throws MBeanException
```

Get the String URI of the default component (or interface) for a particular group

### Returns

*String URI of default version*

### Exceptions

*MBeanException*

---

## getGroup() Method

```
public String getGroup()
```

Get the component group URI for this MBean

### Returns

*String URI of the component group*

---

## isEnabled(String) Method

```
public boolean isEnabled(String componentURI)
    throws VersionException
```

Is the component enabled?

### Parameters

*componentURI*  
String URI of component

### Returns

*component's enabled flag value*

### Exceptions

*VersionException*  
if the versioning manager is disabled or an unexpected error happened.

---

## isVersioningEnabled() Method

```
public boolean isVersioningEnabled()
```

Is Version Manager enabled and initialized?

## Returns

*true or false*

---

## set(String, boolean, long) Method

```
public void set(String componentURI,  
                boolean active,  
                long activationDate)  
    throws VersionException
```

Set both the activation date and the active flag of a particular component. set the deactivation date to default -1 value, which means that the component has no deactivation date)

## Parameters

*componentURI*  
String URI of this component

*active*  
toggle activation value

*activationDate*  
time of activation in seconds since 01.01.1970 00:00.00 GMT

## Exceptions

*VersionException*

---

## set(String, boolean, long, long) Method

```
public void set(String componentURI,  
                boolean active,  
                long activationDate,  
                long deactivationDate)  
    throws VersionException
```

Set both the activation date and the active flag of a particular component.

## Parameters

*componentURI*  
String URI of this component

*active*  
toggle activation value

*activationDate*  
time of activation in seconds since 01.01.1970 00:00.00 GMT

*deactivationDate*

time of deactivation in seconds since 01.01.1970 00:00.00 GMT

## Exceptions

*VersionException*

---

## setActivationTime(String, long) Method

```
public void setActivationTime(String componentURI,  
                             long activationTime)  
    throws VersionException
```

Set the activation time of a particular component , keep the same activation date.

## Parameters

*componentURI*

String URI of component

*activationTime*

time of activation in seconds since 01.01.1970 00:00.00 GMT

## Exceptions

*VersionException*

---

## setDeactivationTime(String, long) Method

```
public void setDeactivationTime(String componentURI,  
                               long deactivationTime)  
    throws VersionException
```

Set the activation time of a particular component , keep the same activation date.

## Parameters

*componentURI*

String URI of component

*deactivationTime*

time of deactivation in seconds since 01.01.1970 00:00.00 GMT

## Exceptions

*VersionException*

---

## setEnabled(String, boolean) Method

```
public void setEnabled(String componentURI,  
                      boolean active)  
    throws VersionException
```

Enable or disable a particular component active attribute

### Parameters

*componentURI*

String URI of this component

*active*

toggle activation value

### Exceptions

*VersionException*

---

## validate() Method

```
public String[] validate()  
    throws VersionException
```

Validate the current Version configuration, keep the same active flag A configuration is valid if there is only one currently active version or if there are multiple active versions and each of them has distinct activation dates.

### Returns

*String array of diagnostics if there are errors or empty array if configuration is valid*

### Exceptions

*VersionException*

# WLIconsoleEventGenConfigurationMBean Interface

*public interface WLIconsoleEventGenConfigurationMBean*

extends ConfigurationMBean

Configuration MBean interface for WLI Console Extension for External Event Generators There is one instance of WLIconsoleEventGenConfigurationMBean per each type of generator. (e.g. MQSeries, database, etc).

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration, NotificationBroadcaster, WebLogicMBean

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```

    public
    String getEventGeneratorTypeName(Locale locale)
        Returns a string name of the type of the event generator in the specified locale, e.g.

    public
    String getWebAppURI()
        Returns a URI to the event generator main page e.g.
  
```

## Methods from interface weblogic.management.configuration.ConfigurationMBean

freezeCurrentValue, getAttributeStringValue, getComments, getNotes, getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes,



`setPersistenceEnabled, touch, unregisterConfigMBean`

#### Methods from interface `javax.management.DynamicMBean`

`getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes`

#### Methods from interface `javax.management.MBeanRegistration`

`postDeregister, postRegister, preDeregister, preRegister`

#### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener, getNotificationInfo, removeNotificationListener`

#### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getEventGeneratorTypeName(Locale) Method

```
public String getEventGeneratorTypeName(Locale locale)
    throws MBeanException
```

Returns a string name of the type of the event generator in the specified locale, e.g. "MQ Series Event Generator"

#### Exceptions

*MBeanException*

---

### getWebAppURI() Method

```
public String getWebAppURI()
    throws MBeanException
```

Returns a URI to the event generator main page e.g. "/xegWebApp/main"

#### Exceptions

*MBeanException*

# WorklistConfigurationMBean Interface

*public interface WorklistConfigurationMBean*

extends ConfigurationMBean

MBean for configuring the worklist

---

## All Superinterfaces

ConfigurationMBean, DynamicMBean, MBeanRegistration,  
NotificationBroadcaster, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Fields from interface weblogic.management.configuration.ConfigurationMBean

CACHING\_STUB\_SVUID

## Method Summary

```
public
String[] getAdminRole()
public
String getTaskCreationRole()
public
String getTaskTrackingLevel()
    The possible values are: "none", "basic", "full".
public
void setAdminRole(String[] adminRoles)
public
void setTaskCreationRole(String taskCreationRole)
public
void setTaskTrackingLevel(String trackingLevel)
```

## Methods from interface weblogic.management.configuration.ConfigurationMBean

freezeCurrentValue, getAttributeStringValue, getComments, getNotes,

getSetFields, getXml, getXmlConverter, isDefaultedMBean, isPersistenceEnabled, preDeregister, registerConfigMBean, restoreDefaultValue, setComments, setDefaultedMBean, setNotes, setPersistenceEnabled, touch, unregisterConfigMBean

### Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### getAdminRole() Method

```
public String[] getAdminRole()
```

---

## getTaskCreationRole() Method

```
public String getTaskCreationRole()
```

---

## getTaskTrackingLevel() Method

```
public String getTaskTrackingLevel()
```

The possible values are: "none", "basic", "full".

---

## setAdminRole(String[]) Method

```
public void setAdminRole(String[] adminRoles)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskCreationRole(String) Method

```
public void setTaskCreationRole(String taskCreationRole)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskTrackingLevel(String) Method

```
public void setTaskTrackingLevel(String trackingLevel)
    throws MBeanException
```

### Exceptions

*MBeanException*

# com.bea.wli.management.runtime Package

## Interface Summary

ArchiverRuntimeMBean	
ChildInstance	information about child processes started by a parent process
EmailEventGenRuntimeMBean	This class provides runtime management for a Email Event Generator
FileEventGeneratorsRuntimeMBean	This class provides a central access point for getting FileEventGenRuntimeMBeans
FileEventGenRuntimeMBean	This class provides runtime management for a File Event Generator
GenericEventGenRuntimeMBean	This class provides runtime management for a XEG Event Generator
HttpEventGenRuntimeMBean	This class provides runtime management for an Http Event Generator.
JmsEventGenRuntimeMBean	This class provides runtime management for a Jms Event Generator
MessageTrackingRuntimeMBean	
MsgBrokerRuntimeMBean	Message broker runtime info
ParentInstance	represents info about parent instance in tracking data for the child process
ProcessInstanceInfo	Represents information about a particular process instance.
ProcessRuntimeMBean	MBean for accessing runtime information about processes.
ProcessStatistics	Represents runtime statistics for a particular process type.
ProcessVariableValue	Represents the value of a process variable on a client that is not running within the server JVM.
SampleRuntimeMBean	
TimerEventGeneratorsRuntimeMBean	This class provides a central access point for getting TimerEventGenRuntimeMBeans
TimerEventGenRuntimeMBean	This class provides runtime management for a Time Event Generator
WorklistRuntimeMBean	MBean for accessing runtime information about the worklist
WorkSubstituteRuntimeMBean	MBean for accessing runtime information about the worklist

## Class Summary

MessageBrokerChannelValue	Represents a message broker channel
MessageBrokerSubscriberValue	Represents a subscriber for a channel
	A workflow with a static and dynamic subscription will have two entries, one for each subscription
ProcessInstanceQuery	Represents a query to obtain information about process instances.
ProcessInstanceQueryResult	Represents the result of a process instance query.
WrappedThrowable	Wraps an exception that was thrown on the server so that information about the exception can be obtained by any client without having to worry about

whether the original exception is in the classpath of the caller.

## Exception Summary

MessageTrackingRuntimeException

# ArchiverRuntimeMBean Interface

*public interface ArchiverRuntimeMBean*

extends RuntimeMBean

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

public  
static  
final long **CACHING\_STUB\_SVUID**

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

public  
String[] **getDataSouceJNDINames()**  
Returns a list of JNDI names for DataSources that can be used for archiving.

public  
long **getEventRowCount()**  
Returns the number of events in the process tracking table.

public  
long **getLastProcessEndTime()**  
Returns a time (in millis) that indicates the last time this process finished.

public  
long **getLastProcessStartTime()**  
Returns a time (in millis) that indicates the last time this process was started.

public  
void **runWLIArchiver()**  
Force the archiver process to run ASAP, instead of waiting for next scheduled time.

public  
void **setLastProcessEndTime(long time)**  
internal

public  
void **setLastProcessStartTime(long time)**  
internal

public  
void **stopCurrentRun()**



Gracefully interrupts the current archiver run.

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.runtime.RuntimeMBean`

`preDeregister`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### **CACHING\_STUB\_SVUID**

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

## getDataSouceJNDINames() Method

```
public String[] getDataSouceJNDINames()  
    throws WLIArchiverException
```

Returns a list of JNDI names for DataSources that can be used for archiving.

### Exceptions

*WLIArchiverException*

---

## getEventRowCount() Method

```
public long getEventRowCount()  
    throws WLIArchiverException
```

Returns the number of events in the process tracking table.

### Exceptions

*WLIArchiverException*

---

## getLastProcessEndTime() Method

```
public long getLastProcessEndTime()  
    throws WLIArchiverException
```

Returns a time (in millis) that indicates the last time this process finished.

### Exceptions

*WLIArchiverException*

---

## getLastProcessStartTime() Method

```
public long getLastProcessStartTime()  
    throws WLIArchiverException
```

Returns a time (in millis) that indicates the last time this process was started.

### Exceptions

*WLIArchiverException*

---

## runWLIArchiver() Method

```
public void runWLIArchiver()  
    throws WLIArchiverException
```

Force the archiver process to run ASAP, instead of waiting for next scheduled time.

### Exceptions

*WLIArchiverException*

---

## setLastProcessEndTime(long) Method

```
public void setLastProcessEndTime(long time)
```

internal

---

## setLastProcessStartTime(long) Method

```
public void setLastProcessStartTime(long time)
```

internal

---

## stopCurrentRun() Method

```
public void stopCurrentRun()  
    throws WLIArchiverException
```

Gracefully interrupts the current archiver run. If the process is not running, this command is ignored.

### Exceptions

*WLIArchiverException*

# ChildInstance Interface

*public interface ChildInstance*

information about child processes started by a parent process

## Method Summary

```
public
    int getActivityID()
public
    int getEventType()
public
    String getProcessInstance()
```

## Method Detail

### **getActivityID() Method**

```
public int getActivityID()
```

#### Returns

*ID of activity in the parent process that caused the child to be invoked*

---

### **getEventType() Method**

```
public int getEventType()
```

#### Returns

*TrackingEventType.EVENT\_TYPE\_CHILD\_PROCESS\_INSTANCE\_START or  
TrackingEventType.EVENT\_TYPE\_CHILD\_PROCESS\_INSTANCE\_CONTINUE*

---

### **getProcessInstance() Method**

```
public String getProcessInstance()
```

#### Returns

*process instance of the child*

# EmailEventGenRuntimeMBean Interface

*public interface* **EmailEventGenRuntimeMBean**

extends `RuntimeMBean`

This class provides runtime management for a Email Event Generator

---

## All Superinterfaces

`DynamicMBean`, `MBeanRegistration`, `NotificationBroadcaster`, `RuntimeMBean`, `WebLogicMBean`

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    long getEmailCount()
        Return the number of emails that have been processed by the event generator.

    public
    long getEmailErrorCount()
        Return the number of errors that have been encountered by the event generator.

    public
    long getErrorCountResetTime()
        Return the timestamp of last error counter reset operation.

    public
    String getEventGeneratorName()
        Get the name of the event generator

    public
    long getResetTime()
        Return the timestamp of last message counter reset operation.

    public
    void resetEmailCount()
        Reset the email count of the event generator.
  
```

```
public resetEmailErrorCount()
    void          Reset the error count of the event generator.
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

```
preDeregister
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getEmailCount() Method

```
public long getEmailCount()
```

Return the number of emails that have been processed by the event generator.

#### Returns

*the number of emails that have been processed by the event generator.*

---

### getEmailErrorCount() Method

```
public long getEmailErrorCount()
```

Return the number of errors that have been encountered by the event generator.

#### Returns

*the number of errors encountered by the event generator.*

---

### getErrorCountResetTime() Method

```
public long getErrorCountResetTime()
```

Return the timestamp of last error counter reset operation.

#### Returns

*the error counter reset timestamp.*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

**getResetTime() Method**

```
public long getResetTime()
```

Return the timestamp of last message counter reset operation.

**Returns**

*the message counter timestamp.*

---

**resetEmailCount() Method**

```
public void resetEmailCount()
```

Reset the email count of the event generator.

---

**resetEmailErrorCount() Method**

```
public void resetEmailErrorCount()
```

Reset the error count of the event generator.



# FileEventGeneratorsRuntimeMBean Interface

*public interface* **FileEventGeneratorsRuntimeMBean**

extends RuntimeMBean

This class provides a central access point for getting FileEventGenRuntimeMBeans

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    FileEventGenRuntimeMBean getRuntimeMBean(String name)
        Get a specific FileEventGenRuntimeMBean

    public
    FileEventGenRuntimeMBean[] getRuntimeMBeans()
        Get a list of all known FileEventGenRuntimeMBean's
  
```

### Methods from interface javax.management.DynamicMBean

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface javax.management.MBeanRegistration

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface javax.management.NotificationBroadcaster

addNotificationListener, getNotificationInfo, removeNotificationListener

**Methods from interface `weblogic.management.runtime.RuntimeMBean`**`preDeregister`**Methods from interface `weblogic.management.WebLogicMBean`**`getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### `getRuntimeMBean(String)` Method

**DEPRECATED** Use `getMBeansByType(FileEventGenRuntimeMBean.TYPE)` and lookup the string

```
public FileEventGenRuntimeMBean getRuntimeMBean(String name)
    throws MBeanException
```

Get a specific `FileEventGenRuntimeMBean`

**Parameters***name*

name of the MBean

**Returns***the specified MBean***Exceptions***MBeanException*

---

**getRuntimeMBeans() Method*****DEPRECATED*** Use getMBeansByType(FileEventGenRuntimeMBean.TYPE);

```
public FileEventGenRuntimeMBean[] getRuntimeMBeans()  
    throws MBeanException
```

Get a list of all known FileEventGenRuntimeMBean's

**Returns***Array of MBean's***Exceptions***MBeanException*

# FileEventGenRuntimeMBean Interface

*public interface FileEventGenRuntimeMBean*

extends RuntimeMBean

This class provides runtime management for a File Event Generator

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    long getErrorCount()
        Get the number of errors that occurred in the event generator

    public
    long getErrorResetTime()
        Get the reset timestamp of the errors for the event generator

    public
    String getEventGeneratorName()
        Get the name of the event generator

    public
    long getFileCount()
        Get the number of files that have been processed by the event generator

    public
    long getResetTime()
        Get the reset timestamp of the event generator

    public
    void resetErrorCount()
        Reset the error count of the event generator
  
```

```
public resetFileCount()
    void          Reset the file count of the event generator
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

```
preDeregister
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getErrorCount() Method

```
public long getErrorCount()
```

Get the number of errors that occurred in the event generator

#### Returns

*the number of errors that occurred in the event generator*

---

### getErrorResetTime() Method

```
public long getErrorResetTime()
```

Get the reset timestamp of the errors for the event generator

#### Returns

*reset timestamp of the errors for the event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### getFileCount() Method

```
public long getFileCount()
```

Get the number of files that have been processed by the event generator

#### Returns

*the number of files that have been processed by the event generator*

---

## **getResetTime() Method**

```
public long getResetTime()
```

Get the reset timestamp of the event generator

### **Returns**

*reset timestamp of the event generator*

---

## **resetErrorCount() Method**

```
public void resetErrorCount()
```

Reset the error count of the event generator

---

## **resetFileCount() Method**

```
public void resetFileCount()
```

Reset the file count of the event generator

# GenericEventGenRuntimeMBean Interface

*public interface GenericEventGenRuntimeMBean*

extends RuntimeMBean

This class provides runtime management for a XEG Event Generator

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

<pre>public static final long</pre>	<pre><b>CACHING_STUB_SVUID</b></pre>	<p>The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean_Stub).</p>
<pre>public static final String</pre>	<pre><b>TYPE</b></pre>	<p>String</p>

## Method Summary

<pre>public long</pre>	<pre><b>getErrorCount()</b></pre>	<p>Get the number of errors that occurred in the event generator</p>
<pre>public long</pre>	<pre><b>getErrorResetTime()</b></pre>	<p>Get the reset timestamp of the errors for the event generator</p>
<pre>public String</pre>	<pre><b>getEventGeneratorName()</b></pre>	<p>Get the name of the event generator</p>
<pre>public long</pre>	<pre><b>getLongProperty(String propertyName)</b></pre>	<p>Returns a long value of a named property (i.e. anything other than message or error count)</p>
<pre>public long</pre>	<pre><b>getMessageCount()</b></pre>	<p>Get the number of messages that have been processed by the event generator</p>
<pre>public long</pre>	<pre><b>getResetTime()</b></pre>	<p>Get the reset timestamp of the event generator</p>



```

public resetErrorCount()
    void          Reset the error count of the event generator
public
    void resetLongProperty(String propertyName)
        Resets a long long value of a named property to 0 (i.e. anything other than message or error
        count)
public
    void resetMessageCount()
        Reset the message count of the event generator

```

#### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

#### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

#### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

`preDeregister`

#### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## TYPE

```
public static final String TYPE
```

## Method Detail

### getErrorCount() Method

```
public long getErrorCount()
```

Get the number of errors that occurred in the event generator

#### Returns

*the number of errors that occurred in the event generator*

---

### getErrorResetTime() Method

```
public long getErrorResetTime()
```

Get the reset timestamp of the errors for the event generator

#### Returns

*reset timestamp of the errors for the event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### getLongProperty(String) Method

```
public long getLongProperty(String propertyName)
```

Returns a long value of a named property (i.e. anything other than message or error count)

---

## getMessageCount() Method

```
public long getMessageCount()
```

Get the number of messages that have been processed by the event generator

### Returns

*the number of files that have been processed by the event generator*

---

## getResetTime() Method

```
public long getResetTime()
```

Get the reset timestamp of the event generator

### Returns

*reset timestamp of the event generator*

---

## resetErrorCount() Method

```
public void resetErrorCount()
```

Reset the error count of the event generator

---

## resetLongProperty(String) Method

```
public void resetLongProperty(String propertyName)
```

Resets a long long value of a named property to 0 (i.e. anything other than message or error count)

---

## resetMessageCount() Method

```
public void resetMessageCount()
```

Reset the message count of the event generator

# HttpEventGenRuntimeMBean Interface

*public interface* **HttpEventGenRuntimeMBean**

extends `RuntimeMBean`

This class provides runtime management for an Http Event Generator.

---

## All Superinterfaces

`DynamicMBean`, `MBeanRegistration`, `NotificationBroadcaster`, `RuntimeMBean`, `WebLogicMBean`

---

## Field Summary

<pre>public static final long</pre>	<pre><b>CACHING_STUB_SVUID</b></pre>	<p>The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean_Stub).</p>
<pre>public static final String</pre>	<pre><b>TYPE</b></pre>	<p>String</p>

## Method Summary

<pre>public long</pre>	<pre><b>getErrorCount()</b></pre>	<p>Return the number of errors that have been encountered by the event generator.</p>
<pre>public long</pre>	<pre><b>getErrorResetTime()</b></pre>	<p>Return the time that the error counter was last reset.</p>
<pre>public String</pre>	<pre><b>getEventGeneratorName()</b></pre>	<p>Return the name of the event generator.</p>
<pre>public long</pre>	<pre><b>getMessageCount()</b></pre>	<p>Get the number of messages that have been processed by the event generator.</p>
<pre>public long</pre>	<pre><b>getResetTime()</b></pre>	<p>Return the time that the message counter was last reset.</p>
<pre>public void</pre>	<pre><b>resetErrorCount()</b></pre>	<p>Reset the error message count of the event generator.</p>

```
public resetMessageCount()
    void          Reset the message count of the event generator.
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

```
preDeregister
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getErrorCount() Method

```
public long getErrorCount()
```

Return the number of errors that have been encountered by the event generator.

#### Returns

*the number of errors that have been encountered by the event generator*

---

### getErrorResetTime() Method

```
public long getErrorResetTime()
```

Return the time that the error counter was last reset.

#### Returns

*the error counter reset timestamp.*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Return the name of the event generator.

#### Returns

*name of the event generator*

---

### getMessageCount() Method

```
public long getMessageCount()
```

Get the number of messages that have been processed by the event generator.

#### Returns

*the number of messages that have been processed by the event generator*

---

**getResetTime() Method**

```
public long getResetTime()
```

Return the time that the message counter was last reset.

**Returns**

*the message counter timestamp*

---

**resetErrorCount() Method**

```
public void resetErrorCount()
```

Reset the error message count of the event generator.

---

**resetMessageCount() Method**

```
public void resetMessageCount()
```

Reset the message count of the event generator.

# JmsEventGenRuntimeMBean Interface

*public interface JmsEventGenRuntimeMBean*

extends RuntimeMBean

This class provides runtime management for a Jms Event Generator

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    String getDestJndiName()
        Return the JNDI name of the Destination.

    public
    String getDestType()
        Return the Destination Type.

    public
    long getErrorCount()
        Return the number of error that have been encountered by the event generator.

    public
    long getErrorResetTime()
        Return the time that the error counter was last reset.

    public
    String getEventGeneratorName()
        Return the name of the event generator

    public
    long getMessageCount()
        Get the number of messages that have been processed by the event generator
  
```



```

public getMessageSelector()
String          Return the message selector string.
public
    long getResetTime()
                Return the time that message counter was last reset.
public
    void resetErrorCount()
                Reset the error message count of the event generator.
public
    void resetMessageCount()
                Reset the message count of the event generator

```

#### Methods from interface `javax.management.DynamicMBean`

```

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes

```

#### Methods from interface `javax.management.MBeanRegistration`

```

postDeregister, postRegister, preDeregister, preRegister

```

#### Methods from interface `javax.management.NotificationBroadcaster`

```

addNotificationListener, getNotificationInfo, removeNotificationListener

```

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

```

preDeregister

```

#### Methods from interface `weblogic.management.WebLogicMBean`

```

getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent

```

## Field Detail

### CACHING\_STUB\_SVUID

```

public static final long CACHING_STUB_SVUID

```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e, "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then

**change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getDestJndiName() Method

```
public String getDestJndiName()
```

Return the JNDI name of the Destination.

#### Returns

*the destination JNDI name.*

---

### getDestType() Method

```
public String getDestType()
```

Return the Destination Type.

#### Returns

*the destination type.*

---

### getErrorCount() Method

```
public long getErrorCount()
```

Return the number of error that have been encountered by the event generator.

#### Returns

*the number of errors that have been encountered by the event generator.*

---

### getErrorResetTime() Method

```
public long getErrorResetTime()
```

Return the time that the error counter was last reset.

**Returns**

*the error counter reset timestamp.*

---

**getEventGeneratorName() Method**

```
public String getEventGeneratorName()
```

Return the name of the event generator

**Returns**

*name of the event generator*

---

**getMessageCount() Method**

```
public long getMessageCount()
```

Get the number of messages that have been processed by the event generator

**Returns**

*the number of messages that have been processed by the event generator*

---

**getMessageSelector() Method**

```
public String getMessageSelector()
```

Return the message selector string.

**Returns**

*the message selector string.*

---

**getResetTime() Method**

```
public long getResetTime()
```

Return the time that message counter was last reset.

**Returns**

*the message counter timestamp.*

---

## **resetErrorCount() Method**

```
public void resetErrorCount()
```

Reset the error message count of the event generator.

---

## **resetMessageCount() Method**

```
public void resetMessageCount()
```

Reset the message count of the event generator

# MessageBrokerChannelValue Class

*public class MessageBrokerChannelValue*

extends Object  
implements Serializable

Represents a message broker channel

---

## Hierarchy

Object  
MessageBrokerChannelValue

## All Implemented Interfaces

Serializable

---

## Field Summary

```
public
static MESSAGE_TYPE_NONE
final String      Message type indicating that the channel cannot be published or subscribed to

public
static MESSAGE_TYPE_RAWDATA
final String      Message type indicating that the channel is for publishing com.bea.data.RawData
                  values

public
static MESSAGE_TYPE_STRING
final String      Message type indicating that the channel is for publishing java.lang.String values

public
static MESSAGE_TYPE_XML
final String      Message type indicating that the channel is for publishing com.bea.xml.XmlObject
                  values
```

## Constructor Summary

*MessageBrokerChannelValue(String channelName, String messageType, int msgcount, int deadMsgCount)*

## Method Summary

```
public
String getChannelName()
    get the channel name

public
    int getDeadMessageCount()
        get the undelivered message
        count

public
    int getMessageCount()
        get the message count

public
String getMessageType()
    get the message type
```

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### **\_MESSAGE\_TYPE\_NONE**

```
public static final String _MESSAGE_TYPE_NONE
```

Message type indicating that the channel cannot be published or subscribed to

---

### **\_MESSAGE\_TYPE\_RAWDATA**

```
public static final String _MESSAGE_TYPE_RAWDATA
```

Message type indicating that the channel is for publishing `com.bea.data.RawData` values

---

### **\_MESSAGE\_TYPE\_STRING**

```
public static final String _MESSAGE_TYPE_STRING
```

Message type indicating that the channel is for publishing `java.lang.String` values

---

### **\_MESSAGE\_TYPE\_XML**

```
public static final String _MESSAGE_TYPE_XML
```

Message type indicating that the channel is for publishing `com.bea.xml.XmlObject` values

## Constructor Detail

### MessageBrokerChannelValue

```
public MessageBrokerChannelValue(String channelName,  
                                String messageType,  
                                int msgcount,  
                                int deadMsgCount)
```

## Method Detail

### getChannelName() Method

```
public String getChannelName()
```

get the channel name

#### Returns

*channel name*

---

### getDeadMessageCount() Method

```
public int getDeadMessageCount()
```

get the undelivered message count

#### Returns

*message count*

---

### getMessageCount() Method

```
public int getMessageCount()
```

get the message count

#### Returns

*message count*

---

## **getMessageType() Method**

```
public String getMessageType()
```

get the message type

### **Returns**

*message type*



# MessageBrokerSubscriberValue Class

*public class MessageBrokerSubscriberValue*

extends Object  
implements Serializable

Represents a subscriber for a channel

A workflow with a static and dynamic subscription will have two entries, one for each subscription

---

## *Hierarchy*

Object  
MessageBrokerSubscriberValue

## *All Implemented Interfaces*

Serializable

---

## Constructor Summary

*MessageBrokerSubscriberValue(String subscriptionRuleName, String subscriberUri, String channelName, String methodName, String controlName, String filterString)*

## Method Summary

```
public
String getChannelName()
    get channel name

public
String getControlName()
    get the control name (may be null if this is a static subscription)

public
String getFilterString()
    get the filter string (may be null if doesn't have a filter)

public
String getMethodName()
    get method name

public
String getSubscriberUri()
```

```

        get subscriber URI
public
String getSubscriptionRuleName()
        get subscriber URI

```

#### Methods from class `java.lang.Object`

```

clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait

```

## Constructor Detail

### MessageBrokerSubscriberValue

```

public MessageBrokerSubscriberValue(String subscriptionRuleName,
                                   String subscriberUri,
                                   String channelName,
                                   String methodName,
                                   String controlName,
                                   String filterString)

```

## Method Detail

### getChannelName() Method

```
public String getChannelName()
```

get channel name

#### Returns

*channel name*

---

### getControlName() Method

```
public String getControlName()
```

get the control name (may be null if this is a static subscription)

#### Returns

*control name*

---

## getFilterString() Method

```
public String getFilterString()
```

get the filter string (may be null if doesn't have a filter)

### Returns

*filter string*

---

## getMethodName() Method

```
public String getMethodName()
```

get method name

### Returns

*method name*

---

## getSubscriberUri() Method

```
public String getSubscriberUri()
```

get subscriber URI

### Returns

*subscriber URI*

---

## getSubscriptionRuleName() Method

```
public String getSubscriptionRuleName()
```

get subscriber URI

### Returns

*subscriber URI*

# MessageTrackingRuntimeException Class

*public class MessageTrackingRuntimeException*

extends Exception

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      MessageTrackingRuntimeException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

*MessageTrackingRuntimeException(String msg, Throwable rootCause)*

*MessageTrackingRuntimeException(String msg)*

## **Method Summary**

### **Methods from java.lang.Throwable**

fillInStackTrace, getCause, getLocalizedMessage, getMessage,  
getStackTrace, initCause, printStackTrace, printStackTrace,  
printStackTrace, setStackTrace, toString

### **Methods from class java.lang.Object**

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### MessageTrackingRuntimeException

```
public MessageTrackingRuntimeException(String msg,  
                                     Throwable rootCause)
```

---

### MessageTrackingRuntimeException

```
public MessageTrackingRuntimeException(String msg)
```

# MessageTrackingRuntimeMBean Interface

*public interface MessageTrackingRuntimeMBean*

extends RuntimeMBean

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

```
public
    int getConversationCount(Date startTime, Date endTime)
        Gets conversation count.

public
    int getConversationCountByPartner(String partnerId, Date startTime, Date endTime)
        Gets conversation count by partner.

public
    int getConversationCountByPartners(String fromPartnerId, String toPartnerId, Date
        startTime, Date endTime)
        Gets conversation count by partners.

public
    int getMessageCountByConversationAndDirection(String conversationId, String
        direction, Date startTime, Date endTime)
        Gets message count by conversation and direction.

public
    int getMessageCountByDirection(String direction, Date startTime, Date endTime)
        Gets message count by direction.

public
    int getMessageCountByPartnerAndDirection(String partnerId, String direction,
        Date startTime, Date endTime)
        Gets message count by partner and direction.

public
    int
```

***getMessageCountByPartnersAndDirection(String fromPartnerId, String toPartnerId, String direction, Date startTime, Date endTime)***

Gets message count by partners and direction.

**Methods from interface `javax.management.DynamicMBean`**

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

**Methods from interface `javax.management.MBeanRegistration`**

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

**Methods from interface `javax.management.NotificationBroadcaster`**

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

**Methods from interface `weblogic.management.runtime.RuntimeMBean`**

`preDeregister`

**Methods from interface `weblogic.management.WebLogicMBean`**

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### getConversationCount(Date, Date) Method

```
public int getConversationCount(Date startTime,
                               Date endTime)
    throws MessageTrackingRuntimeException
```

Gets conversation count.

#### Parameters

*startTime*  
the start timestamp of the time period interested in

*endTime*  
the end timestamp of the time period interested in

#### Returns

*total conversations that occurred within the passed time period*

#### Exceptions

*MessageTrackingRuntimeException*

---

### getConversationCountByPartner(String, Date, Date) Method

```
public int getConversationCountByPartner(String partnerId,
                                          Date startTime,
                                          Date endTime)
    throws MessageTrackingRuntimeException
```

Gets conversation count by partner.

#### Parameters

*partnerId*  
the Id of the partner interested in

*startTime*  
the start timestamp of the time period interested in

*endTime*  
the end timestamp of the time period interested in

#### Returns

*total conversations that occurred with the the specified partner*



**Exceptions***MessageTrackingRuntimeException*

---

**getConversationCountByPartners(String, String, Date, Date) Method**

```
public int getConversationCountByPartners(String fromPartnerId,
                                         String toPartnerId,
                                         Date startTime,
                                         Date endTime)
    throws MessageTrackingRuntimeException
```

Gets conversation count by partners.

**Parameters**

*fromPartnerId*  
the Id of the from partner

*toPartnerId*  
the Id of the to partner

*startTime*  
the start timestamp of the time period interested in

*endTime*  
the end timestamp of the time period interested in

**Returns**

*total conversations that occurred between the specified partners*

**Exceptions***MessageTrackingRuntimeException*

---

**getMessageCountByConversationAndDirection(String, String, Date, Date) Method**

```
public int getMessageCountByConversationAndDirection(String conversationId,
                                                      String direction,
                                                      Date startTime,
                                                      Date endTime)
    throws MessageTrackingRuntimeException
```

Gets message count by conversation and direction.

**Parameters***conversationId*

the Id of the conversation interested in

*direction*

"sent" or "received"

*startTime*

the start timestamp of the time period interested in

*endTime*

the end timestamp of the time period interested in

**Returns***total messages exchanged in a specified conversation and in the specified direction***Exceptions***MessageTrackingRuntimeException*

---

**getMessageCountByDirection(String, Date, Date) Method**

```

public int getMessageCountByDirection(String direction,
                                     Date startTime,
                                     Date endTime)
    throws MessageTrackingRuntimeException

```

Gets message count by direction.

**Parameters***direction*

"sent" or "received"

*startTime*

the start timestamp of the time period interested in

*endTime*

the end timestamp of the time period interested in

**Returns***total messages exchanged in the specified direction***Exceptions***MessageTrackingRuntimeException*

---

**getMessageCountByPartnerAndDirection(String, String, Date, Date) Method**

```
public int getMessageCountByPartnerAndDirection(String partnerId,
                                                String direction,
                                                Date startTime,
                                                Date endTime)
    throws MessageTrackingRuntimeException
```

Gets message count by partner and direction.

**Parameters**

*partnerId*  
the Id of the partner interested in

*direction*  
"sent" or "received"

*startTime*  
the start timestamp of the time period interested in

*endTime*  
the end timestamp of the time period interested in

**Returns**

*total messages exchanged by the specified partner in the specified direction*

**Exceptions**

*MessageTrackingRuntimeException*

---

**getMessageCountByPartnersAndDirection(String, String, String, Date, Date) Method**

```
public int getMessageCountByPartnersAndDirection(String fromPartnerId,
                                                  String toPartnerId,
                                                  String direction,
                                                  Date startTime,
                                                  Date endTime)
    throws MessageTrackingRuntimeException
```

Gets message count by partners and direction.

**Parameters**

*fromPartnerId*  
the Id of the from partner

*toPartnerId*  
the Id of the to partner

*direction*

"sent" or "received"

*startTime*

the start timestamp of the time period interested in

*endTime*

the end timestamp of the time period interested in

## **Returns**

*total messages exchanged between the specified partners in the specified direction*

## **Exceptions**

*MessageTrackingRuntimeException*

# MsgBrokerRuntimeMBean Interface

*public interface* **MsgBrokerRuntimeMBean**

extends RuntimeMBean

Message broker runtime info

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```
public
static CACHING_STUB_SVUID
final long
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

```
public
MessageBrokerChannelValue[] getChannelInfo()
    get the list of channels

public
MessageBrokerChannelValue getChannelInfoByName(String name)
    get the list of channels

public Date
getResetDate()
    The date of the last reset

public
MessageBrokerSubscriberValue[] getSubscriptionInfo()
    get the list of subscribers

public
MessageBrokerSubscriberValue[] getSubscriptionInfoByChannelName(String
channelName)
    get the list of subscribers

public void
resetMsgCount()
    Resets the count of processed messages to zero

public void
resetMsgCount(String channelName)
```

Resets the count of processed messages to zero for a specific channel

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.runtime.RuntimeMBean`

`preDeregister`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

## getChannelInfo() Method

```
public MessageBrokerChannelValue[] getChannelInfo()  
    throws MBeanException
```

get the list of channels

### Returns

*list of channels*

### Exceptions

*MBeanException*

---

## getChannelInfoByName(String) Method

```
public MessageBrokerChannelValue getChannelInfoByName(String name)  
    throws MBeanException
```

get the list of channels

### Returns

*list of channels*

### Exceptions

*MBeanException*

---

## getResetDate() Method

```
public Date getResetDate()  
    throws MBeanException
```

The date of the last reset

### Returns

*date*

### Exceptions

*MBeanException*

---

## getSubscriptionInfo() Method

```
public MessageBrokerSubscriberValue[] getSubscriptionInfo()  
    throws MBeanException
```

get the list of subscribers

### Returns

*list of subscribers*

### Exceptions

*MBeanException*

---

## getSubscriptionInfoByChannelName(String) Method

```
public MessageBrokerSubscriberValue[] getSubscriptionInfoByChannelName(String channelName)  
    throws MBeanException
```

get the list of subscribers

### Returns

*list of subscribers*

### Exceptions

*MBeanException*

---

## resetMsgCount() Method

```
public void resetMsgCount()  
    throws MBeanException
```

Resets the count of processed messages to zero

### Exceptions

*MBeanException*

---

## resetMsgCount(String) Method

```
public void resetMsgCount(String channelName)  
    throws MBeanException
```

Resets the count of processed messages to zero for a specific channel



## Exceptions

*MBeanException*

# ParentInstance Interface

*public interface **ParentInstance***

represents info about parent instance in tracking data for the child process

## Method Summary

```
public
    int getEventType()
public
    String getProcessInstance()
```

## Method Detail

### **getEventType() Method**

```
public int getEventType()
```

#### **Returns**

*actionType **TrackingEventType.EVENT\_TYPE\_PARENT\_PROCESS\_INSTANCE\_START** or  
**TrackingEventType.EVENT\_TYPE\_PARENT\_PROCESS\_INSTANCE\_CONTINUE***

---

### **getProcessInstance() Method**

```
public String getProcessInstance()
```

#### **Returns**

*process instance of the parent*

# ProcessInstanceInfo Interface

*public interface* **ProcessInstanceInfo**

extends `Serializable`

Represents information about a particular process instance.

## Related Topics

{ @link `ProcessRuntimeMBean`}

## All Superinterfaces

`Serializable`

## Method Summary

```

public String
    getDisplayName()
        Returns the display name for the process.

public long
    getExecTime()
        Returns the total elapsed time in terms of milliseconds.

public String
    getId()
        Returns the conversation id of this process instance

public String
    getLabel()
        Returns the process label for this process instance.

public Date
    getLastStatusChangeTime()
        Returns the time when the status of this process has changed.

public
ProcessNodeInfo[] getPendingActivities()
        Returns an array of pending activities, namely, nodes where the instance is
        blocked.

public
WrappedThrowable[] getProcessExceptions()
        Returns a list of fatal exception for the instance.

public String
    getServiceURI()
        Returns the service URI for the process type

public long

```

```

        getSLA()
            Returns the SLA value in milliseconds for this process.

    public long
        getSLAWarning()
            Returns the SLA Warning threshold in milliseconds.

    public double
        getSLAWarningThreshold()
            Returns the SLA Warning threshold for this process as a fraction of the
            actual SLA.

    public Date
        getStartTime()
            Returns the starting time for this process instance.

    public
    ProcessStatus getStatus()
            Returns the current status of this instance.

    public boolean
        isDetailed()
            Returns true if the detailed information about this instance is available.

```

## Method Detail

### getDisplayName() Method

```
public String getDisplayName()
```

Returns the display name for the process. This information can be obtained via `ProcessConfigurationMBean.getDisplayName()` as well.

---

### getExecTime() Method

```
public long getExecTime()
```

Returns the total elapsed time in terms of milliseconds. This figure is the wall-clock execution time, and does not exclude the time an instance spent waiting for input. Elapsed time is computed differently depending on the current status of the instances. For running, suspended, and frozen instances the execution time is simply the difference between the current time and the start time. For aborted, terminated and completed instances it is the difference between the last status change time (i.e., whenever the instance has been aborted, terminated, or completed) and the start time.

---

### getId() Method

```
public String getId()
```

Returns the conversation id of this process instance

---

## getLabel() Method

```
public String getLabel()
```

Returns the process label for this process instance. For instances that are no longer running, this returns the last process label that was set by the instance.

---

## getLastStatusChangeTime() Method

```
public Date getLastStatusChangeTime()
```

Returns the time when the status of this process has changed. The return value can be used to determine when a process has completed, terminated, suspended, aborted, and frozen. This method should be used in conjunction with `ProcessInstanceInfo.getStatus()` method.

---

## getPendingActivities() Method

```
public ProcessNodeInfo[] getPendingActivities()
```

Returns an array of pending activities, namely, nodes where the instance is blocked. A null return value indicates that the information is unavailable.

### Related Topics

```
{ @link ProcessRuntimeMBean#getPendingActivities}  
{ @link #isDetailed}.
```

---

## getProcessExceptions() Method

```
public WrappedThrowable[] getProcessExceptions()
```

Returns a list of fatal exception for the instance. Usually just one.

---

## getServiceURI() Method

```
public String getServiceURI()
```

Returns the service URI for the process type

---

## getSLA() Method

```
public long getSLA()
```

Returns the SLA value in milliseconds for this process. This value can also be obtained via `ProcessConfigurationMBean.getSLA()`, and is included here for convenience.

## getSLAWarning() Method

```
public long getSLAWarning()
```

Returns the SLA Warning threshold in milliseconds. This is equal to `getSLA( ) * getSLAWarningThreshold`

---

## getSLAWarningThreshold() Method

```
public double getSLAWarningThreshold()
```

Returns the SLA Warning threshold for this process as a fraction of the actual SLA. This value can also be obtained via `ProcessConfigurationMBean.getSLAWarningThreshold( )`, and is included here for convenience

---

## getStartTime() Method

```
public Date getStartTime()
```

Returns the starting time for this process instance. Returns null if there was an error obtaining the start time.

---

## getStatus() Method

```
public ProcessStatus getStatus()
```

Returns the current status of this instance.

---

## isDetailed() Method

```
public boolean isDetailed()
```

Returns true if the detailed information about this instance is available. When false, pending activities and process exceptions are unavailable. In other words the calls to methods `ProcessInstanceInfo.getPendingActivities( )`, and `ProcessInstanceInfo.getProcessExceptions( )` will return null. Detail information is available only when obtaining information about one particular process instance, rather than querying for more than one instance.

## Related Topics

```
ProcessRuntimeMBean.getProcessInstanceInfo(String, boolean)
```

# ProcessInstanceQuery Class

*public class* **ProcessInstanceQuery**

extends `Object`  
implements `Serializable`

Represents a query to obtain information about process instances.

## Related Topics

`ProcessRuntimeMBean`

---

## Hierarchy

`Object`  
`ProcessInstanceQuery`

## All Implemented Interfaces

`Serializable`

---

## Constructor Summary

### *ProcessInstanceQuery()*

Constructor for creating an empty query.

### *ProcessInstanceQuery(String serviceURI, ProcessStatus status)*

Convenience constructor for querying instances of the given service that are in the given state.

*ProcessInstanceQuery(String serviceURI, ProcessStatus status, long startTimeMin, long startTimeMax, long statusChangeTimeMin, long statusChangeTimeMax, long elapsedMin, long elapsedMax, String label, boolean aboveSLA, boolean aboveSLAWarning, int count)*

### *ProcessInstanceQuery(String instanceId)*

Constructor for querying a single process instance

## Method Summary

```

    public
    boolean equals(Object o)

    public int
        getCount()

    public long
        getElapsedMax()

    public long
        getElapsedMin()

    public
    String getInstanceId()
        Returns the instance (conversation) id component of the query.

    public
    String getLabel()

    public
    String getServiceURI()
        Returns the serviceURI to filter on.

    public long
        getStartTimeMax()

    public long
        getStartTimeMin()

    public
    ProcessStatus[] getStatus()
        Returns the array ProcessStatus object to filter on.

    public long
        getStatusChangeTimeMax()

    public long
        getStatusChangeTimeMin()

    public int
        hashCode()

    public
    boolean isAboveSLA()

    public
    boolean isAboveSLAWarning()

    public
    boolean isSingleInstanceQuery()
        Returns true if the query filters only on the instance (coversation) id.

    public void
        setAboveSLA(boolean aboveSLA)
        Allows filtering instances based on their SLA status.

    public void
        setAboveSLAWarning(boolean aboveSLAWarning)
        Allows filtering instances based on their SLA Warning Threshold status.

    public void

```



## Java Class Reference

```
setCount(int count)
    Sets the maximum number of results to return.

public void setElapsedMax(long elapsedMax)
    Sets the maximum elapsed time for the instances that are returned.

public void setElapsedMin(long elapsedMin)
    Sets the minimum elapsed time for the instances that are returned.

public void setInstanceId(String instanceId)
    Sets the instance (conversation) id component of the query.

public void setLabel(String label)
    Adds a predicate to the query so that only instances whose process label
    containing this value are returned.

public void setServiceURI(String serviceURI)
    Sets the serviceURI predicate so that only instances of a particular process are
    returned.

public void setStartTimeMax(long startTimeMax)
    Adds a predicate to the query so that only instances that started before the given
    time are returned.

public void setStartTimeMin(long startTimeMin)
    Adds a predicate to the query so that only instances that started after the given
    time are returned.

public void setStatus(ProcessStatus[] status)
    Sets the process status predicate so that only instances that are in one of the states
    are returned.

public void setStatus(ProcessStatus status)
    Convenience method for query instances that are in the given state.

public void setStatusChangeTimeMax(long statusChangeTimeMax)
    Adds a predicate to the query so that only instances that has aborted, completed,
    or terminated before the given time are returned.

public void setStatusChangeTimeMin(long statusChangeTimeMin)
    Adds a predicate to the query so that only instances that has aborted, completed,
    or terminated after the given time are returned.

public
String toString()
```

### Methods from class java.lang.Object

clone, finalize, getClass, notify, notifyAll, wait, wait, wait

## Constructor Detail

### ProcessInstanceQuery

```
public ProcessInstanceQuery()
```

Constructor for creating an empty query. An empty query returns all the instances

---

### ProcessInstanceQuery

```
public ProcessInstanceQuery(String serviceURI,
                           ProcessStatus status)
```

Convenience constructor for querying instances of the given service that are in the given state.

---

### ProcessInstanceQuery

```
public ProcessInstanceQuery(String serviceURI,
                           ProcessStatus[] status,
                           long startTimeMin,
                           long startTimeMax,
                           long statusChangeTimeMin,
                           long statusChangeTimeMax,
                           long elapsedMin,
                           long elapsedMax,
                           String label,
                           boolean aboveSLA,
                           boolean aboveSLAWarning,
                           int count)
```

---

### ProcessInstanceQuery

```
public ProcessInstanceQuery(String instanceId)
```

Constructor for querying a single process instance

## Method Detail

### equals(Object) Method

```
public boolean equals(Object o)
```

*Overrides*

Object.equals(Object)

---

**getCount() Method**

```
public int getCount()
```

---

**getElapsedMax() Method**

```
public long getElapsedMax()
```

---

**getElapsedMin() Method**

```
public long getElapsedMin()
```

---

**getInstanceId() Method**

```
public String getInstanceId()
```

Returns the instance (conversation) id component of the query. If instance id is set all other predicates are ignored.

---

**getLabel() Method**

```
public String getLabel()
```

---

**getServiceURI() Method**

```
public String getServiceURI()
```

Returns the serviceURI to filter on. Returns null if the query does not filter based on service URI.

---

**getStartTimeMax() Method**

```
public long getStartTimeMax()
```

---

**getStartTimeMin() Method**

```
public long getStartTimeMin()
```

---

**getStatus() Method**

```
public ProcessStatus[] getStatus()
```

Returns the array `ProcessStatus` object to filter on. Returns null if the query does not filter based on the process status.

---

### **getStatusChangeTimeMax() Method**

```
public long getStatusChangeTimeMax()
```

---

### **getStatusChangeTimeMin() Method**

```
public long getStatusChangeTimeMin()
```

---

### **hashCode() Method**

```
public int hashCode()
```

#### *Overrides*

```
Object.hashCode()
```

---

### **isAboveSLA() Method**

```
public boolean isAboveSLA()
```

---

### **isAboveSLAWarning() Method**

```
public boolean isAboveSLAWarning()
```

---

### **isSingleInstanceQuery() Method**

```
public boolean isSingleInstanceQuery()
```

Returns true if the query filters only on the instance (conversation) id.

---

### **setAboveSLA(boolean) Method**

```
public void setAboveSLA(boolean aboveSLA)
```

Allows filtering instances based on their SLA status. A true value causes instances that exceeded their SLAs to be returned. default value for this component is `false`. This method can be used in conjunction with the `ProcessInstanceQuery.setAboveSLAWarning(boolean)` to return:

- all instances that exceeded their SLAs but not SLA Warning Thresholds.
- all instances that exceeded their SLAs or SLA Warning Thresholds.
- all instances that exceeded their SLA Warning Threshold but not SLA

## setAboveSLAWarning(boolean) Method

```
public void setAboveSLAWarning(boolean aboveSLAWarning)
```

Allows filtering instances based on their SLA Warning Threshold status. A true value causes instances that exceeded their SLA Warning Thresholds to be returned. default value for this component is false. This method can be used in conjunction with the `ProcessInstanceQuery.setAboveSLA(boolean)` to return:

- all instances that exceeded their SLAs but not SLA Warning Thresholds.
  - all instances that exceeded their SLAs or SLA Warning Thresholds.
  - all instances that exceeded their SLA Warning Threshold but not SLA
- 

## setCount(int) Method

```
public void setCount(int count)
```

Sets the maximum number of results to return.

---

## setElapsedMax(long) Method

```
public void setElapsedMax(long elapsedMax)
```

Sets the maximum elapsed time for the instances that are returned. The default value for this component is `Long.MAX_VALUE`

### Parameters

*elapsedMax*  
maximum elapsed time in milliseconds

---

## setElapsedMin(long) Method

```
public void setElapsedMin(long elapsedMin)
```

Sets the minimum elapsed time for the instances that are returned. The default value for this component is 0

### Parameters

*elapsedMin*  
minimum elapsed time in milliseconds

---

## setInstanceId(String) Method

```
public void setInstanceId(String instanceId)
```

Sets the instance (conversation) id component of the query. This resets all other predicates that may have been set previously.

---

## setLabel(String) Method

```
public void setLabel(String label)
```

Adds a predicate to the query so that only instances whose process label **containing** this value are returned.

---

## setServiceURI(String) Method

```
public void setServiceURI(String serviceURI)
```

Sets the serviceURI predicate so that only instances of a particular process are returned. A null value is allowed and indicates all processes, effectively disabling the filtering on serviceURI.

---

## setStartTimeMax(long) Method

```
public void setStartTimeMax(long startTimeMax)
```

Adds a predicate to the query so that only instances that started before the given time are returned. A value of Long.MAX\_VALUE effectively disabled this predicate. The default value for this component is Long.MAX\_VALUE

### Parameters

*startTimeMax*  
time in milliseconds

---

## setStartTimeMin(long) Method

```
public void setStartTimeMin(long startTimeMin)
```

Adds a predicate to the query so that only instances that started after the given time are returned. A value of 0 effectively disables this predicate. The default value for this component is 0

### Parameters

*startTimeMin*  
time in milliseconds

---

## setStatus(ProcessStatus[]) Method

```
public void setStatus(ProcessStatus[] status)
```

Sets the process status predicate so that only instances that are in one of the states are returned. A null value or an empty array indicates that process status is not considered when querying instances.

---

## setStatus(ProcessStatus) Method

```
public void setStatus(ProcessStatus status)
```

Convenience method for query instances that are in the given state.

---

## setStatusChangeTimeMax(long) Method

```
public void setStatusChangeTimeMax(long statusChangeTimeMax)
```

Adds a predicate to the query so that only instances that has aborted, completed, or terminated before the given time are returned. A value of Long.MAX\_VALUE effectively disables this predicate. The default value for this component is Long.MAX\_VALUE

---

## setStatusChangeTimeMin(long) Method

```
public void setStatusChangeTimeMin(long statusChangeTimeMin)
```

Adds a predicate to the query so that only instances that has aborted, completed, or terminated after the given time are returned. A value of 0 effectively disables this predicate. The default value for this component is 0

### Parameters

*statusChangeTimeMin*  
time in milliseconds

---

## toString() Method

```
public String toString()
```

### Overrides

Object.toString()

# ProcessInstanceQueryResult Class

*public class ProcessInstanceQueryResult*

extends Object  
implements Serializable

Represents the result of a process instance query.

## Related Topics

{ @link ProcessInstanceQuery}  
{ @link ProcessRuntimeMBean}

---

## Hierarchy

Object  
ProcessInstanceQueryResult

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*ProcessInstanceQueryResult(String instanceIds, Date queryDate, boolean isAll)*

## Method Summary

```
public
String getInstanceId(int index)
    Convenience method for obtaining a particular instance id in the result.

public
String[] getInstanceIds()
    Returns conversation ids of all the instances that satisfied the relevant
    ProcessInstanceQuery

public
Date getQueryDate()
    Returns the approximate date when this result was obtained.

public
int getResultCard()
```



Simply returns the number of instances ids returned.

```
public  
boolean isAll()
```

Returns true if the all the instances that satisfied the original `ProcessInstanceQuery` was returned.

### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### ProcessInstanceQueryResult

```
public ProcessInstanceQueryResult(String[] instanceIds,  
                                   Date queryDate,  
                                   boolean isAll)
```

## Method Detail

### getInstanceId(int) Method

```
public String getInstanceId(int index)
```

Convenience method for obtaining a particular instance id in the result.

---

### getInstanceIds() Method

```
public String[] getInstanceIds()
```

Returns conversation ids of all the instances that satisfied the relevant `ProcessInstanceQuery`

---

### getQueryDate() Method

```
public Date getQueryDate()
```

Returns the **approximate** date when this result was obtained.

---

### getResultCard() Method

```
public int getResultCard()
```

Simply returns the number of instances ids returned. Returns the same value as `getInstanceIds().length`

---

### **isAll() Method**

```
public boolean isAll()
```

Returns true if the all the instances that satisfied the original `ProcessInstanceQuery` was returned. Some instances will not be returned if the query limited the result cardinality explicitly (via `ProcessInstanceQuery.setCount(int)`) or the result cardinality was larger than the server-specific upper limit.

# ProcessRuntimeMBean Interface

*public interface* **ProcessRuntimeMBean**

extends `RuntimeMBean`

MBean for accessing runtime information about processes. There is one `ProcessRuntimeMBean` instance for all processes.

---

## All Superinterfaces

`DynamicMBean`, `MBeanRegistration`, `NotificationBroadcaster`, `RuntimeMBean`, `WebLogicMBean`

---

## Field Summary

```
public
static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

```
public ChildInstance[]
    getChildProcessDataFromTracking(String
        parentProcessInstance)
    Returns information on child process instances called by
    given process instance

public int
    getCount(ProcessInstanceQuery query)
    Returns the number of instances that satisfy the given query.

public
ProcessVariableValue getInitialMessage(String serviceURI, String instanceId)
    Returns the message which started a process, or null if the
    process was not started by an incoming message.

public String
    getLabel(String instanceId)
    Returns the current value of a process's label.

public ParentInstance[]
    getParentProcessDataFromTracking(String
        childProcessInstance)
    Returns information on parent process instance that called the
    given process instance
```

## Java Class Reference

```
public ProcessNodeInfo[] getPendingActivities(String serviceURI, String
instanceId)
    Returns an array of nodes where an instance is currently
    blocked

public WrappedThrowable[]
getProcessExceptions(String processURI, String
instanceID)
    Returns a list of fatal exception for the instance.

public ProcessInstanceInfo
getProcessInstanceInfo(String id, boolean
includeDetails)
    Returns information about a process instance.

    public List
getProcessInstanceInfo(String[] ids)
        Returns information about the given process instances

    public
ProcessInstanceQueryResult getProcessInstances(ProcessInstanceQuery query)
    Returns ids of process instances that satisfy the given query
    embedded in the ProcessInstanceQueryResult
    object.

    public
ProcessVariableValue getProcessVariable(String serviceURI, String
instanceId, String variableName)
    Gets the current value of a process variable

public ProcessStatistics
getStatistics(String serviceURI, long minStartTime, long
maxStartTime, boolean onlyActiveInstances)
    Returns statistics for a process type.

public ProcessStatistics[]
getStatistics()
    Returns statistics for all process types for which there is at
    least one record in the summary table.

    public ProcessStatus
getStatus(String instanceId)
    Returns the status of an instance.

    public Map
getVariableSummary(String serviceURI, String
instanceId)
    Returns summary values for process variables.

public void
resume(String serviceURI, String instanceId)
    Resumes a suspended process instance

public void
suspend(String serviceURI, String instanceId)
    Suspends a process instance

public void
terminate(String serviceURI, String instanceId)
    Terminates a process instance
```

## Java Class Reference

```
public void unfreeze(String serviceURI, String instanceId)  
    Unfreezes a frozen instance
```

### Methods from interface `javax.management.DynamicMBean`

`getAttribute`, `getAttributes`, `getMBeanInfo`, `invoke`, `setAttribute`, `setAttributes`

### Methods from interface `javax.management.MBeanRegistration`

`postDeregister`, `postRegister`, `preDeregister`, `preRegister`

### Methods from interface `javax.management.NotificationBroadcaster`

`addNotificationListener`, `getNotificationInfo`, `removeNotificationListener`

### Methods from interface `weblogic.management.runtime.RuntimeMBean`

`preDeregister`

### Methods from interface `weblogic.management.WebLogicMBean`

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`, `isCachingDisabled`, `isRegistered`, `setName`, `setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

## getChildProcessDataFromTracking(String) Method

```
public ChildInstance[] getChildProcessDataFromTracking(String parentProcessInstance)
    throws MBeanException
```

Returns information on child process instances called by given process instance

### Exceptions

*MBeanException*

---

## getCount(ProcessInstanceQuery) Method

```
public int getCount(ProcessInstanceQuery query)
    throws MBeanException
```

Returns the number of instances that satisfy the given query. If the query is too expensive to execute or the number of instances is unknown, simply returns -1.

### Exceptions

*MBeanException*

---

## getInitialMessage(String, String) Method

```
public ProcessVariableValue getInitialMessage(String serviceURI,
    String instanceId)
    throws MBeanException, ConversationNotFoundException, ProcessNotFoundException, ProcessIns
```

Returns the message which started a process, or null if the process was not started by an incoming message.

### Parameters

*serviceURI*  
service URI of the process

*instanceId*  
id of the process instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessNotFoundException*  
*ProcessInstanceNotFoundException*

---

## getLabel(String) Method

```
public String getLabel(String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessInstanceNotFoundException
```

Returns the current value of a process's label.

### Parameters

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessInstanceNotFoundException*

---

## getParentProcessDataFromTracking(String) Method

```
public ParentInstance[] getParentProcessDataFromTracking(String childProcessInstance)  
    throws MBeanException
```

Returns information on parent process instance that called the given process instance

### Exceptions

*MBeanException*

---

## getPendingActivities(String, String) Method

```
public ProcessNodeInfo[] getPendingActivities(String serviceURI,  
                                              String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessNotFoundException, ProcessIns
```

Returns an array of nodes where an instance is currently blocked

### Parameters

*serviceURI*  
service URI for the process

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*

*ProcessNotFoundException*  
*ProcessInstanceNotFoundException*

---

## getProcessExceptions(String, String) Method

```
public WrappedThrowable[] getProcessExceptions(String processURI,  
                                              String instanceID)  
    throws MBeanException
```

Returns a list of fatal exception for the instance. Usually just one.

### Exceptions

*MBeanException*

---

## getProcessInstanceInfo(String, boolean) Method

```
public ProcessInstanceInfo getProcessInstanceInfo(String id,  
                                                  boolean includeDetails)  
    throws MBeanException
```

Returns information about a process instance.

### Parameters

*id*  
the id of the process instance.

*includeDetails*  
whether to include detailed information in the result. The detail information consists of variable values, pending activities, and initial message.

### Returns

*information about the given instance. Returns null if no information could be found for the given instance*

### Exceptions

*MBeanException*

---

## getProcessInstanceInfo(String[]) Method

```
public List getProcessInstanceInfo(String[] ids)  
    throws MBeanException
```

Returns information about the given process instances



**Parameters***ids*

an array of process instance ids

**Returns**

*a List of ProcessInstanceInfo objects. The result could have less element than the number of instance ids. This happens If no information could be found for an instance.*

**Exceptions***MBeanException***getProcessInstances(ProcessInstanceQuery) Method**

```
public ProcessInstanceQueryResult getProcessInstances(ProcessInstanceQuery query)
    throws MBeanException
```

Returns ids of process instances that satisfy the given query embedded in the ProcessInstanceQueryResult object.

**Exceptions***MBeanException***getProcessVariable(String, String, String) Method**

```
public ProcessVariableValue getProcessVariable(String serviceURI,
                                              String instanceId,
                                              String variableName)
    throws MBeanException, ConversationNotFoundException, NoSuchFieldException, ProcessInstanceNotFoundException
```

Gets the current value of a process variable

**Parameters***serviceURI*

service URI of the process

*instanceId*

the id of the instance

**Exceptions***MBeanException**ConversationNotFoundException**NoSuchFieldException**ProcessInstanceNotFoundException*

## getStatistics(String, long, long, boolean) Method

```
public ProcessStatistics getStatistics(String serviceURI,  
                                     long minStartTime,  
                                     long maxStartTime,  
                                     boolean onlyActiveInstances)  
  
    throws MBeanException
```

Returns statistics for a process type.

### Parameters

*serviceURI*

the service URI for the process

*minStartTime*

include only instances that have started after this time. Use 0 if this is not applicable to the filtering criteria.

*maxStartTime*

include only instances that have started before this time. Use `Long.MAX_VALUE` if this is not applicable of the filtering criteria.

*onlyActiveInstances*

include only active instances, namely instances that are running, suspended, or frozen. if `false` all instances are considered.

### Exceptions

*MBeanException*

---

## getStatistics() Method

```
public ProcessStatistics[] getStatistics()  
    throws MBeanException
```

Returns statistics for all process types for which there is at least one record in the summary table. If there is no record of an instance of a particular process type (perhaps because no instance of that process has ever run since the last archive and purge process) then no statistics is returned for that process type.

### Returns

*an array of ProcessStatistics objects one for each process type.*

**Exceptions***MBeanException*

---

**getStatus(String) Method**

```
public ProcessStatus getStatus(String instanceId)
    throws MBeanException
```

Returns the status of an instance.

**Parameters**

*instanceId*  
the id of the instance

**Exceptions***MBeanException*

---

**getVariableSummary(String, String) Method**

```
public Map getVariableSummary(String serviceURI,
                               String instanceId)
    throws ProcessInstanceNotFoundException, ProcessNotFoundException, MBeanException
```

Returns summary values for process variables. The result is a map from variable names (String) to ProcessVariableValue contains a short string representation of the value.

**Parameters**

*serviceURI*  
service URI of the process

*instanceId*  
id of the instance

**Exceptions**

*ProcessInstanceNotFoundException*  
when the given instance was not found

*ProcessNotFoundException*  
when the given process is not found

*MBeanException*  
any other exception is wrapped with a ManagementException

---

## resume(String, String) Method

```
public void resume(String serviceURI,  
                  String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessNotFoundException, ProcessIns
```

Resumes a suspended process instance

### Parameters

*serviceURI*  
service URI for the process

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessNotFoundException*  
*ProcessInstanceNotFoundException*

---

## suspend(String, String) Method

```
public void suspend(String serviceURI,  
                   String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessInstanceNotFoundException, Pr
```

Suspends a process instance

### Parameters

*serviceURI*  
service URI for the process

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessInstanceNotFoundException*  
*ProcessNotFoundException*

---

## terminate(String, String) Method

```
public void terminate(String serviceURI,  
                      String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessNotFoundException, ProcessIns
```

Terminates a process instance

### Parameters

*serviceURI*  
service URI for the process

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessNotFoundException*  
*ProcessInstanceNotFoundException*

---

## unfreeze(String, String) Method

```
public void unfreeze(String serviceURI,  
                     String instanceId)  
    throws MBeanException, ConversationNotFoundException, ProcessNotFoundException, ProcessIns
```

Unfreezes a frozen instance

### Parameters

*serviceURI*  
service URI for the process

*instanceId*  
the id of the instance

### Exceptions

*MBeanException*  
*ConversationNotFoundException*  
*ProcessNotFoundException*  
*ProcessInstanceNotFoundException*

# ProcessStatistics Interface

*public interface* **ProcessStatistics**

extends `Serializable`

Represents runtime statistics for a particular process type.

## Related Topics

{ @link `ProcessRuntimeMBean#getStatistics`}

---

## All Superinterfaces

`Serializable`

---

## Method Summary

```
public
double getAvgExecTime()
    Returns the average execution time of the instances that are selected.

public
    int getNumAborted()
        Returns the number of aborted instances of the process represented by this statistics
        object.

public
    int getNumAll()
        Returns the number of all the instances for which these statistics are returned.

public
    int getNumCompleted()
        Returns the number of successfully completed instances of the process represented by this
        statistics object.

public
    int getNumExceededSLA()
        Returns the number of instances that exceeded the SLA set for this process.

public
    int getNumFrozen()
        Returns the number of frozen instances of the process represented by this statistics object.

public
    int getNumRunning()
        Returns the number of running instances of the process represented by this statistics
        object.

public
    int getNumSuspended()
```

Returns the number of suspended instances of the process represented by this statistics object.

```
public
    int getNumTerminated()
```

Returns the number of terminated instances of the process represented by this statistics object.

```
public
    String getServiceURI()
```

Returns the service URI of the process for which the statistics are computed

## Method Detail

### getAvgExecTime() Method

```
public double getAvgExecTime()
```

Returns the average execution time of the instances that are selected.

---

### getNumAborted() Method

```
public int getNumAborted()
```

Returns the number of aborted instances of the process represented by this statistics object.

---

### getNumAll() Method

```
public int getNumAll()
```

Returns the number of all the instances for which these statistics are returned. This is merely a sum of running, suspended, aborted, frozen, completed, and terminated instances, and is here for only as convenience.

---

### getNumCompleted() Method

```
public int getNumCompleted()
```

Returns the number of successfully completed instances of the process represented by this statistics object.

---

### getNumExceededSLA() Method

```
public int getNumExceededSLA()
```

Returns the number of instances that exceeded the SLA set for this process. If there is no SLA set for the process a value of 0 is returned.

---

**getNumFrozen() Method**

```
public int getNumFrozen()
```

Returns the number of frozen instances of the process represented by this statistics object.

---

**getNumRunning() Method**

```
public int getNumRunning()
```

Returns the number of running instances of the process represented by this statistics object.

---

**getNumSuspended() Method**

```
public int getNumSuspended()
```

Returns the number of suspended instances of the process represented by this statistics object.

---

**getNumTerminated() Method**

```
public int getNumTerminated()
```

Returns the number of terminated instances of the process represented by this statistics object.

---

**getServiceURI() Method**

```
public String getServiceURI()
```

Returns the service URI of the process for which the statistics are computed



# ProcessVariableValue Interface

*public interface* **ProcessVariableValue**

extends `Serializable`

Represents the value of a process variable on a client that is not running within the server JVM. This representation is needed because the client who is requesting the values of variables may not have all the classes in its classpath, and simply returning a value would cause a `ClassNotFoundException`. This class attempts to solve this problem by providing a representation for values whose class may or may not be known by the client. This class is also used to pass a short representation of a variable value.

---

## All Superinterfaces

`Serializable`

---

## Method Summary

```

public
Object deserializeValue()
    Convenience method for constructing the value object from its serialized form.

public
String getDeclaredType()
    Returns the declared type of the variable.

public
String getName()
    Returns the name of the process variable whose value is represented

public
String getOriginalInstanceType()
    Returns the original type of the value

public
Object getValueRepresentation()
    Returns a representation of the value of the process variable.

public
boolean isObjectValueRep()
    Returns true if the value of the variable obtained via
    ProcessVariableValue.getValueRepresentation() is some form of the
    original value, rather than serialized value.

public
boolean isSerializedValueRep()
    Returns true if the value of the variable is represented in its serialized form.

public
boolean isShortRepresentation()

```

Returns true if this object contains a short String based representation of the variable value.

```
public
boolean isTooLarge()
```

Returns true if the object was too large.

```
public
boolean isUnrepresentableValue()
```

Returns true if the value of the variable cannot be represented outside the server JVM, and therefore is unavailable.

```
public
boolean isXMLType()
```

Returns if the declared type is an XML type.

## Method Detail

### deserializeValue() Method

```
public Object deserializeValue()
    throws IOException, ClassNotFoundException
```

Convenience method for constructing the value object from its serialized form. This method should only be called if `ProcessVariableValue.isSerializedValueRep( )` returns true

### Exceptions

*IOException*

*ClassNotFoundException*

if the class for the object being represented is not found.

---

### getDeclaredType() Method

```
public String getDeclaredType()
```

Returns the declared type of the variable.

---

### getName() Method

```
public String getName()
```

Returns the name of the process variable whose value is represented

---

### getOriginalInstanceType() Method

```
public String getOriginalInstanceType()
```

Returns the original type of the value

---

## getValueRepresentation() Method

```
public Object getValueRepresentation()
```

Returns a representation of the value of the process variable. Depending on the type(class) of the process variable one of the following values is returned.

- if this object represents the summary value of a variable the result is always String and the value of String is obtained by calling the toString method on the original value. The value will be truncated if it is large. The size is system dependent. However short representation of XML values and RawData is always null. This is done in order to avoid materializing potentially huge amounts of data.
  - otherwise if the variable is a primitive type the return type will be the class type of the primitive type. e.g., a value of type Integer will be returned if the variable type is int.
  - otherwise if the variable is a String, or a class type of a primitive type (e.g., Integer then the original value is returned.
  - otherwise if the variable is an XML type then an untyped XmlObject is returned, regardless of the type of the xml.
  - otherwise if the variable is an instance of RawData, the result is a byte array containing the bytes of the raw data.
  - otherwise if the variable is an instance of throwable then the type of the return value will be the same as the original throwable value provided that it is on clients classpath. If the throwable is not on clients classpath then an approximate representation of the original throwable value is returned. See WrappedThrowable for more information about simple form of throwables.
  - Otherwise If the variable can be serialized without errors the result will be the byte array representing the serialized value. See ProcessVariableValue.deserializeValue( ) for obtaining the original value.
  - Otherwise the variable is not representable on the client and a null value is returned. Note that a null value may indicate a proper value rather than representability of a value. Consult ProcessVariableValue.isUnrepresentableValue( ) before using the value. Always use ProcessVariableValue.isTooLarge( ), ProcessVariableValue.isObjectValueRep( ), ProcessVariableValue.isSerializedValueRep( ), ProcessVariableValue.isUnrepresentableValue( ) to interpret the return value of this method.
- 

## isObjectValueRep() Method

```
public boolean isObjectValueRep()
```

Returns true if the value of the variable obtained via ProcessVariableValue.getValueRepresentation( ) is some form of the original value, rather than serialized value.

---

## isSerializedValueRep() Method

```
public boolean isSerializedValueRep()
```

Returns true if the value of the variable is represented in its serialized form. This implies that ProcessVariableValue.getValueRepresentation( ) returns the serialized form in a byte

array. You should first try to get the deserialized (original) Object via `ProcessVariableValue.deserializeValue()`.

---

### **isShortRepresentation() Method**

```
public boolean isShortRepresentation()
```

Returns true if this object contains a short String based representation of the variable value. If so the value returned by `ProcessVariableValue.getValueRepresentation()` will always be a String (possibly null) subject to size limitations.

---

### **isTooLarge() Method**

```
public boolean isTooLarge()
```

Returns true if the object was too large. If so the value returned by `ProcessVariableValue.getValueRepresentation()` will always be null.

---

### **isUnrepresentableValue() Method**

```
public boolean isUnrepresentableValue()
```

Returns true if the value of the variable cannot be represented outside the server JVM, and therefore is unavailable.

---

### **isXMLType() Method**

```
public boolean isXMLType()
```

Returns if the declared type is an XML type.

# SampleRuntimeMBean Interface

*public interface SampleRuntimeMBean*

extends RuntimeMBean

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Method Summary

```
public
    int doBar(String b)
public
    String getFoo()
public
    List getList()
public
    void setFoo(String f)
```

### Methods from interface javax.management.DynamicMBean

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface javax.management.MBeanRegistration

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface javax.management.NotificationBroadcaster

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface weblogic.management.runtime.RuntimeMBean

preDeregister

### Methods from interface weblogic.management.WebLogicMBean

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Method Detail

### doBar(String) Method

```
public int doBar(String b)
```

---

### getFoo() Method

```
public String getFoo()
```

---

### getList() Method

```
public List getList()
```

---

### setFoo(String) Method

```
public void setFoo(String f)
```

# TimerEventGeneratorsRuntimeMBean Interface

*public interface* **TimerEventGeneratorsRuntimeMBean**

extends RuntimeMBean

This class provides a central access point for getting TimerEventGenRuntimeMBeans

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    TimerEventGenRuntimeMBean getRuntimeMBean(String name)
        Get a specific TimerEventGenRuntimeMBean

    public
    TimerEventGenRuntimeMBean[] getRuntimeMBeans()
        Get a list of all known TimerEventGenRuntimeMBean's
  
```

### Methods from interface javax.management.DynamicMBean

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface javax.management.MBeanRegistration

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface javax.management.NotificationBroadcaster

addNotificationListener, getNotificationInfo, removeNotificationListener

**Methods from interface `weblogic.management.runtime.RuntimeMBean`**`preDeregister`**Methods from interface `weblogic.management.WebLogicMBean`**`getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent`

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getRuntimeMBean(String) Method

**DEPRECATED** Use `getMBeansByType(TimerEventGenRuntimeMBean.TYPE)` and lookup the string

```
public TimerEventGenRuntimeMBean getRuntimeMBean(String name)
    throws MBeanException
```

Get a specific `TimerEventGenRuntimeMBean`



**Parameters***name*

name of the MBean

**Returns***the specified MBean***Exceptions***MBeanException*

---

**getRuntimeMBeans() Method*****DEPRECATED*** Use getMBeansByType(TimerEventGenRuntimeMBean.TYPE);

```
public TimerEventGenRuntimeMBean[] getRuntimeMBeans()  
    throws MBeanException
```

Get a list of all known TimerEventGenRuntimeMBean's

**Returns***Array of MBean's***Exceptions***MBeanException*

# TimerEventGenRuntimeMBean Interface

*public interface* **TimerEventGenRuntimeMBean**

extends RuntimeMBean

This class provides runtime management for a Time Event Generator

---

## All Superinterfaces

DynamicMBean, MBeanRegistration, NotificationBroadcaster, RuntimeMBean, WebLogicMBean

---

## Field Summary

```

    public
    static CACHING_STUB_SVUID
    final long
        The magic number that is used to generate the serialVersionUID of the generated
        MBean Stub (xxxMBean_Stub).

    public
    static TYPE
    final String
        String
  
```

## Method Summary

```

    public
    long getErrorCount()
        Get the number of errors that occurred in the event generator

    public
    long getErrorResetTime()
        Get the reset timestamp of the errors for the event generator

    public
    String getEventGeneratorName()
        Get the name of the event generator

    public
    long getResetTime()
        Get the reset timestamp of the event generator

    public
    long getTimerCount()
        Get the number of timer events that have been processed by the event generator

    public
    void resetErrorCount()
        Reset the error count of the event generator
  
```

```
public resetTimerCount()
    void          Reset the timer count of the event generator
```

#### Methods from interface `javax.management.DynamicMBean`

```
getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute,
setAttributes
```

#### Methods from interface `javax.management.MBeanRegistration`

```
postDeregister, postRegister, preDeregister, preRegister
```

#### Methods from interface `javax.management.NotificationBroadcaster`

```
addNotificationListener, getNotificationInfo, removeNotificationListener
```

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

```
preDeregister
```

#### Methods from interface `weblogic.management.WebLogicMBean`

```
getMBeanInfo, getName, getObjectNames, getParent, getType,
isCachingDisabled, isRegistered, setName, setParent
```

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

---

## TYPE

```
public static final String TYPE
```

## Method Detail

### getErrorCount() Method

```
public long getErrorCount()
```

Get the number of errors that occurred in the event generator

#### Returns

*the number of errors that occurred in the event generator*

---

### getErrorResetTime() Method

```
public long getErrorResetTime()
```

Get the reset timestamp of the errors for the event generator

#### Returns

*reset timestamp of the errors for the event generator*

---

### getEventGeneratorName() Method

```
public String getEventGeneratorName()
```

Get the name of the event generator

#### Returns

*name of the event generator*

---

### getResetTime() Method

```
public long getResetTime()
```

Get the reset timestamp of the event generator

#### Returns

*reset timestamp of the event generator*

---

## **getTimerCount() Method**

```
public long getTimerCount()
```

Get the number of timer events that have been processed by the event generator

### **Returns**

*the number of timer events that have been processed*

---

## **resetErrorCount() Method**

```
public void resetErrorCount()
```

Reset the error count of the event generator

---

## **resetTimerCount() Method**

```
public void resetTimerCount()
```

Reset the timer count of the event generator

# WorklistRuntimeMBean Interface

*public interface* **WorklistRuntimeMBean**

extends `RuntimeMBean`

MBean for accessing runtime information about the worklist

---

## All Superinterfaces

`DynamicMBean`, `MBeanRegistration`, `NotificationBroadcaster`, `RuntimeMBean`, `WebLogicMBean`

---

## Field Summary

public  
static  
final long **CACHING\_STUB\_SVUID**  
The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

public  
void **abortTask**(*String taskId*)

public  
void **archiveTasks**()

public  
void **assignToUser**(*String userName, String taskId*)

public  
*String* **assignToUserInGroup**(*String groupName, String taskId*)

public  
void **assignToUsersAndGroups**(*String[] userNamesAndOrGroupNames, String taskId*)

public  
void **claimTask**(*String taskId*)

public  
void **claimTask**(*String user, String taskId*)

public  
void **completeTask**(*String taskId*)

public  
void **deleteTask**(*String taskId*)

```

    public deleteTasks(TaskSelector selector)
        void
public int
    getSelectedTasksNumber(TaskSelector selector)
    public
String[] getTaskIds(TaskSelector selector)
    public
TaskInfo getTaskInfo(String taskId)
    public
TaskInfo[] getTaskInfos(TaskSelector selector)
    public
String[] getTaskProperties(String[] propertyNames, String taskId)
    public
String[] getTaskPropertiesNames(String taskId)
        Return all the task properties names
    public
String getTaskProperty(String propertyName, String taskId)
        get a task Property
    public
TaskMessage getTaskRequest(String taskId)
    public
XmlObject getTaskRequestAsXmlObject(String taskId)
    public
TaskMessage getTaskResponse(String taskId)
    public
XmlObject getTaskResponseAsXmlObject(String taskId)
    public
void purgeTasks()
    public
String[] removeTaskProperties(String[] propertyNames, String taskId)
        Return the values of the removed properties, the value is null if not found
    public
void resumeTask(String taskId)
    public
void returnTask(String taskId)
    public
void setTaskCanBeAborted(Boolean value, String taskId)
    public
void setTaskCanBeReassigned(Boolean value, String taskId)
    public
void setTaskCanBeReturned(Boolean value, String taskId)
    public
void setTaskClaimDueBusinessDate(BusinessTime time, String taskId)
    public
void setTaskClaimDueDate(Date claimDueDate, String taskId)

```

## Java Class Reference

```
public setTaskComment(String comment, String taskId)
    void
public
    void setTaskCompletionDueBusinessDate(BusinessTime time, String taskId)
public
    void setTaskCompletionDueDate(Date completeDueDate, String taskId)
public
    void setTaskOwner(String owner, String taskId)
public
    void setTaskPriority(Integer priority, String taskId)
public
String[] setTaskProperties(String[] propertyNames, String[] values, String taskId)
public
String setTaskProperty(String propertyName, String value, String taskId)
    Return the previous value if there was one
public
    void setTaskRequest(TaskMessage request, String taskId)
public
    void setTaskRequestAsXmlObject(XmlObject request, String mimeType, String
        taskId)
public
    void setTaskResponse(TaskMessage response, String taskId)
public
    void setTaskResponseAsXmlObject(XmlObject response, String mimeType,
        String taskId)
public
    void startTask(String taskId)
public
    void stopTask(String taskId)
public
    void suspendTask(String taskId)
```

### Methods from interface javax.management.DynamicMBean

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

### Methods from interface javax.management.MBeanRegistration

postDeregister, postRegister, preDeregister, preRegister

### Methods from interface javax.management.NotificationBroadcaster

addNotificationListener, getNotificationInfo, removeNotificationListener

### Methods from interface weblogic.management.runtime.RuntimeMBean

preDeregister



**Methods from interface `weblogic.management.WebLogicMBean`**

`getMBeanInfo`, `getName`, `getObjectNames`, `getParent`, `getType`,  
`isCachingDisabled`, `isRegistered`, `setName`, `setParent`

**Field Detail****CACHING\_STUB\_SVUID**

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the `serialVersionUID` of the generated MBean Stub (`xxxMBean_Stub`). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean `isEnabled()`"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

**Method Detail****`abortTask(String)` Method**

```
public void abortTask(String taskId)
    throws MBeanException
```

**Exceptions**

*MBeanException*

---

**`archiveTasks()` Method**

```
public void archiveTasks()
    throws MBeanException
```

**Exceptions**

*MBeanException*

---

## assignToUser(String, String) Method

```
public void assignToUser(String userName,
                        String taskId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## assignToUserInGroup(String, String) Method

```
public String assignToUserInGroup(String groupName,
                                String taskId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## assignToUsersAndGroups(String[], String) Method

```
public void assignToUsersAndGroups(String[] userNamesAndOrGroupNames,
                                String taskId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## claimTask(String) Method

```
public void claimTask(String taskId)
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## claimTask(String, String) Method

```
public void claimTask(String user,
                    String taskId)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### completeTask(String) Method

```
public void completeTask(String taskId)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### deleteTask(String) Method

```
public void deleteTask(String taskId)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### deleteTasks(TaskSelector) Method

```
public void deleteTasks(TaskSelector selector)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getSelectedTasksNumber(TaskSelector) Method

```
public int getSelectedTasksNumber(TaskSelector selector)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getTaskIds(TaskSelector) Method

```
public String[] getTaskIds(TaskSelector selector)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getTaskInfo(String) Method

```
public TaskInfo getTaskInfo(String taskId)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getTaskInfos(TaskSelector) Method

```
public TaskInfo[] getTaskInfos(TaskSelector selector)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getTaskProperties(String[], String) Method

```
public String[] getTaskProperties(String[] propertyNames,
                                   String taskId)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getTaskPropertiesNames(String) Method

```
public String[] getTaskPropertiesNames(String taskId)
    throws MBeanException
```

Return all the task properties names

## Exceptions

*MBeanException*

---

## getTaskProperty(String, String) Method

```
public String getTaskProperty(String propertyName,  
                             String taskId)  
    throws MBeanException
```

get a task Property

### Exceptions

*MBeanException*

---

## getTaskRequest(String) Method

```
public TaskMessage getTaskRequest(String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## getTaskRequestAsXmlObject(String) Method

```
public XmlObject getTaskRequestAsXmlObject(String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## getTaskResponse(String) Method

```
public TaskMessage getTaskResponse(String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## getTaskResponseAsXmlObject(String) Method

```
public XmlObject getTaskResponseAsXmlObject(String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## purgeTasks() Method

```
public void purgeTasks()  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## removeTaskProperties(String[], String) Method

```
public String[] removeTaskProperties(String[] propertyNames,  
                                     String taskId)  
    throws MBeanException
```

Return the values of the removed properties, the value is null if not found

## Exceptions

*MBeanException*

---

## resumeTask(String) Method

```
public void resumeTask(String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## returnTask(String) Method

```
public void returnTask(String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## setTaskCanBeAborted(Boolean, String) Method

```
public void setTaskCanBeAborted(Boolean value,  
                                String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskCanBeReassigned(Boolean, String) Method

```
public void setTaskCanBeReassigned(Boolean value,  
                                   String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskCanBeReturned(Boolean, String) Method

```
public void setTaskCanBeReturned(Boolean value,  
                                  String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskClaimDueBusinessDate(BusinessTime, String) Method

```
public void setTaskClaimDueBusinessDate(BusinessTime time,  
                                         String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskClaimDueDate(Date, String) Method

```
public void setTaskClaimDueDate(Date claimDueDate,  
                                 String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### setTaskComment(String, String) Method

```
public void setTaskComment(String comment,  
                           String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### setTaskCompletionDueBusinessDate(BusinessTime, String) Method

```
public void setTaskCompletionDueBusinessDate(BusinessTime time,  
                                             String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### setTaskCompletionDueDate(Date, String) Method

```
public void setTaskCompletionDueDate(Date completeDueDate,  
                                     String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### setTaskOwner(String, String) Method

```
public void setTaskOwner(String owner,  
                         String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---



## setTaskPriority(Integer, String) Method

```
public void setTaskPriority(Integer priority,  
                           String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskProperties(String[], String[], String) Method

```
public String[] setTaskProperties(String[] propertyNames,  
                                String[] values,  
                                String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskProperty(String, String, String) Method

```
public String setTaskProperty(String propertyName,  
                              String value,  
                              String taskId)  
    throws MBeanException
```

Return the previous value if there was one

### Exceptions

*MBeanException*

---

## setTaskRequest(TaskMessage, String) Method

```
public void setTaskRequest(TaskMessage request,  
                           String taskId)  
    throws MBeanException
```

### Exceptions

*MBeanException*

---

## setTaskRequestAsXmlObject(XmlObject, String, String) Method

```
public void setTaskRequestAsXmlObject(XmlObject request,  
                                      String mimeType,
```

`String taskId)`  
`throws MBeanException`

## Exceptions

*MBeanException*

---

## setTaskResponse(TaskMessage, String) Method

```
public void setTaskResponse(TaskMessage response,  
                             String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## setTaskResponseAsXmlObject(XmlObject, String, String) Method

```
public void setTaskResponseAsXmlObject(XmlObject response,  
                                         String mimeType,  
                                         String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## startTask(String) Method

```
public void startTask(String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## stopTask(String) Method

```
public void stopTask(String taskId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

## **suspendTask(String) Method**

```
public void suspendTask(String taskId)
    throws MBeanException
```

### **Exceptions**

*MBeanException*

# WorkSubstituteRuntimeMBean Interface

*public interface* **WorkSubstituteRuntimeMBean**

extends `RuntimeMBean`

MBean for accessing runtime information about the worklist

---

## All Superinterfaces

`DynamicMBean`, `MBeanRegistration`, `NotificationBroadcaster`, `RuntimeMBean`, `WebLogicMBean`

---

## Field Summary

```
public
static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub).

## Method Summary

```
public String[]
    addWorkSubstituteRules(WorkSubstituteRule[]
        workSubstituteRules)

public
WorkSubstituteRule[] getAllWorkSubstituteRules()

public String
    getWorkSubstitute(String user, Date date)

public
WorkSubstituteRule getWorkSubstituteRule(String ruleId)

public
WorkSubstituteRule[] getWorkSubstituteRulesWithSource(String source)

public
WorkSubstituteRule[] getWorkSubstituteRulesWithTarget(String target)

public void
    removeWorkSubstituteRules(String[] workSubstituteRuleIds)

public void
    updateWorkSubstituteRules(WorkSubstituteRule[] rules)
```

Methods from interface `javax.management.DynamicMBean`

getAttribute, getAttributes, getMBeanInfo, invoke, setAttribute, setAttributes

#### Methods from interface `javax.management.MBeanRegistration`

postDeregister, postRegister, preDeregister, preRegister

#### Methods from interface `javax.management.NotificationBroadcaster`

addNotificationListener, getNotificationInfo, removeNotificationListener

#### Methods from interface `weblogic.management.runtime.RuntimeMBean`

preDeregister

#### Methods from interface `weblogic.management.WebLogicMBean`

getMBeanInfo, getName, getObjectNames, getParent, getType, isCachingDisabled, isRegistered, setName, setParent

## Field Detail

### CACHING\_STUB\_SVUID

```
public static final long CACHING_STUB_SVUID
```

The magic number that is used to generate the serialVersionUID of the generated MBean Stub (xxxMBean\_Stub). You must change this value whenever you make one of the changes to this interface:

- remove an existing method.
- change the signature of an existing "getter" or "setter" method.
- change the signature of an existing "boolean getter", i.e. "boolean isEnabled()"

Otherwise you don't need to change this value. For instance you can add new methods, or you can change the signature of methods that don't start with "get", "set" or "is". If you are in doubt then **change** the SVUID. It is a good idea to increment this value when changing.

## Method Detail

### addWorkSubstituteRules(WorkSubstituteRule[]) Method

```
public String[] addWorkSubstituteRules(WorkSubstituteRule[] workSubstituteRules)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getAllWorkSubstituteRules() Method

```
public WorkSubstituteRule[] getAllWorkSubstituteRules()  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getWorkSubstitute(String, Date) Method

```
public String getWorkSubstitute(String user,  
                                Date date)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getWorkSubstituteRule(String) Method

```
public WorkSubstituteRule getWorkSubstituteRule(String ruleId)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getWorkSubstituteRulesWithSource(String) Method

```
public WorkSubstituteRule[] getWorkSubstituteRulesWithSource(String source)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### getWorkSubstituteRulesWithTarget(String) Method

```
public WorkSubstituteRule[] getWorkSubstituteRulesWithTarget(String target)  
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### **removeWorkSubstituteRules(String[]) Method**

```
public void removeWorkSubstituteRules(String[] workSubstituteRuleIds)
    throws MBeanException
```

## Exceptions

*MBeanException*

---

### **updateWorkSubstituteRules(WorkSubstituteRule[]) Method**

```
public void updateWorkSubstituteRules(WorkSubstituteRule[] rules)
    throws MBeanException
```

## Exceptions

*MBeanException*

# WrappedThrowable Class

*public class* **WrappedThrowable**

extends `Object`  
implements `Serializable`

Wraps an exception that was thrown on the server so that information about the exception can be obtained by any client without having to worry about whether the original exception is in the classpath of the caller. This is required if an exception thrown on the server is not necessarily on clients classpath. Such Throwables are wrapped by serializing them into a byte array so that they can be selectively deserialized on the client. If the deserialization of the original exception fails the client can ask for a simpler representation of the original exception that is guaranteed to be on clients classpath. This simple representation has stack trace and exception message of the original throwable

---

## *Hierarchy*

`Object`  
`WrappedThrowable`

## *All Implemented Interfaces*

`Serializable`

---

## Constructor Summary

**WrappedThrowable**(*Throwable wrappedThrowable, int maxDataSize*)

Constructor for wrapping a throwable.

## Method Summary

`public`  
`Throwable` **getOriginalForm**()  
Returns the original (wrapped) throwable.

`public`  
`String` **getOriginalThrowableClassName**()  
Returns the class name of the wrapped Throwable.

`public`  
`Throwable` **getSimpleForm**()  
Returns a simple representation of the original throwable.

`public`



Throwable **getThrowable()**

Returns the original exception if it can be deserialized without errors and if it is not null.

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### WrappedThrowable

```
public WrappedThrowable(Throwable wrappedThrowable,
                        int maxSize)
```

Constructor for wrapping a throwable.

## Method Detail

### getOriginalForm() Method

```
public Throwable getOriginalForm()
    throws IOException, ClassNotFoundException
```

Returns the original (wrapped) throwable. This method should be called only if the class of the wrapped throwable is known to be in the classpath of the caller. The caller should typically call `WrappedThrowable.getSimpleForm()` to get the alternative simpler representation of this class if the original form could not be obtained due to an exception

#### Returns

*the original wrapped throwable. Returns null if the original Throwable could not be serialized due to some serialization error.*

#### Exceptions

*IOException*  
*ClassNotFoundException*

---

### getOriginalThrowableClassName() Method

```
public String getOriginalThrowableClassName()
```

Returns the class name of the wrapped Throwable. This method can be used to verify if the original throwable is in the classpath of the caller, before calling `WrappedThrowable.getOriginalForm()`

## getSimpleForm() Method

```
public Throwable getSimpleForm()
```

Returns a simple representation of the original throwable. This method should be used if the call to `WrappedThrowable.getOriginalForm()` has failed because the class of the original `Throwable` was not on callers classpath.

### Returns

*a simple representation of the original throwable.*

---

## getThrowable() Method

```
public Throwable getThrowable()
```

Returns the original exception if it can be deserialized without errors and if it is not null. Otherwise returns the simple form.

# **com.bea.wli.user.api Package**

## **Interface Summary**

UserInfo

## **Exception Summary**

UserInfoException

# UserInfo Interface

*public interface UserInfo*

## Method Summary

```
public
    void deleteWLIUserInfo(String userid)
        Removes this user information from the WLI database

public
    String getUserCalendarId(String userid)
        Retrieve the calendar id for this user

public
    String getUserEmail(String userid)
        Retrieve the email address for this user.

public
    boolean groupExistsWLS(String groupId)

public
    void sync()
        Removes all users from the WLI database that do not exist in
        WLS

public
    void unsetUserCalendar(String userid)
        Unsets the calendar for this user.

public
    void updateUserCalendar(String userid, String calendarId)
        Sets the calendar for this user.

public
    void updateUserEmail(String userid, String email)
        Sets the email address for this user.
```

## Method Detail

### deleteWLIUserInfo(String) Method

```
public void deleteWLIUserInfo(String userid)
    throws UserInfoException, RemoteException
```

Removes this user information from the WLI database

### Exceptions

*UserInfoException*  
*RemoteException*

## getUserCalendarId(String) Method

```
public String getUserCalendarId(String userid)
    throws UserInfoException, RemoteException
```

Retrieve the calendar id for this user

### Returns

*The calendar id for the user or null if there is no calendar.*

### Exceptions

*UserInfoException*  
*RemoteException*

---

## getUserEmail(String) Method

```
public String getUserEmail(String userid)
    throws UserInfoException, RemoteException
```

Retrieve the email address for this user.

### Returns

*The email address of the user or null if there is no email address is found.*

### Exceptions

*UserInfoException*  
*RemoteException*

---

## groupExistsWLS(String) Method

```
public boolean groupExistsWLS(String groupId)
    throws UserInfoException, RemoteException
```

### Exceptions

*UserInfoException*  
*RemoteException*

---

## sync() Method

```
public void sync()
    throws UserInfoException, RemoteException
```

Removes all users from the WLI database that do not exist in WLS

### Exceptions

*UserInfoException*  
*RemoteException*

---

### unsetUserCalendar(String) Method

```
public void unsetUserCalendar(String userid)
    throws UserInfoException, RemoteException
```

Unsets the calendar for this user.

### Exceptions

*UserInfoException*  
*RemoteException*

---

### updateUserCalendar(String, String) Method

```
public void updateUserCalendar(String userid,
                               String calendarId)
    throws UserInfoException, RemoteException
```

Sets the calendar for this user. A user record is created if it does not exist in the WLI database. Exception is thrown if the calendar does not exist or the user is not a valid WLS user.

### Exceptions

*UserInfoException*  
*RemoteException*

---

### updateUserEmail(String, String) Method

```
public void updateUserEmail(String userid,
                             String email)
    throws UserInfoException, RemoteException
```

Sets the email address for this user. A user record is created if it does not exist in the WLI database. Exception is thrown if the user is not a valid WLS user.

### Exceptions

*UserInfoException*  
*RemoteException*

# UserInfoException Class

*public class UserInfoException*

extends Exception

---

## Hierarchy

Object  
  Throwable  
    Exception  
      UserInfoException

## All Implemented Interfaces

Serializable

---

## Constructor Summary

*UserInfoException(String message, Throwable cause)*

*UserInfoException(String message)*

## Method Summary

public  
String **getMessage()**

### Methods from java.lang.Throwable

fillInStackTrace, getCause, getLocalizedMessage, getStackTrace,  
initCause, printStackTrace, printStackTrace, printStackTrace,  
setStackTrace, toString

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### UserInfoException

```
public UserInfoException(String message,  
                        Throwable cause)
```

---

### UserInfoException

```
public UserInfoException(String message)
```

## Method Detail

### getMessage() Method

```
public String getMessage()
```

#### *Overrides*

```
    Throwable.getMessage()
```



# com.bea.wli.worklist.api Package

## Interface Summary

WorklistManager

WorkSubstituteManager The work substitue allows to manage the work substitute rules.

## Class Summary

ActionType	The action type class describes all the actions possible on the worklist.
Assignee	This class provide information about the task assignee.
BusinessTime	This class is used to store a business time.
StateType	This calss describes the different possible states of a task.
TaskCreation	This class provide information for creatin a task.
TaskInfo	This class provide information about the runtime state of a Task.
TaskMessage	This class describes a task message composed of: – a byte array that is the message itself.
TaskSelector	TaskSelector is used to select tasks based on specific properties.
WorkSubstituteRule	This class provide information about a workSubstituteRule.

## Exception Summary

AssignmentException Exception thrown when an assignment operation fails.

ManagementException Exception thrown when an operation attempted on the Worklist Manager fails.

# ActionType Class

*public class ActionType*

extends Object  
implements Serializable

The action type class describes all the actions possible on the worklist. You can refer to this class to determine what is the action name stored in the archiving tables.

---

## Hierarchy

```

Object
  ActionType
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Field Summary

```

    public
    static ABORT
final ActionType ActionType

    public
    static ADD_LISTENER
final ActionType ActionType

    public
    static ASSIGN
final ActionType ActionType

    public
    static CLAIM
final ActionType ActionType

    public
    static CLAIM_EXPIRE
final ActionType ActionType

    public
    static COMPLETE
final ActionType ActionType

    public
    static CREATE
final ActionType ActionType

    public
    static EXPIRE
final ActionType
  
```

```

        ActionType

        public
        static REMOVE_LISTENER
final ActionType ActionType

        public
        static RESUME
final ActionType ActionType

        public
        static RETURN
final ActionType ActionType

        public
        static START
final ActionType ActionType

        public
        static STOP
final ActionType ActionType

        public
        static SUSPEND
final ActionType ActionType

        public
        static UPDATE_CAN_BE_ABORTED
final ActionType ActionType

        public
        static UPDATE_CAN_BE_REASSIGNED
final ActionType ActionType

        public
        static UPDATE_CAN_BE_RETURNED
final ActionType ActionType

        public
        static UPDATE_CLAIM_DATE
final ActionType ActionType

        public
        static UPDATE_COMMENT
final ActionType ActionType

        public
        static UPDATE_EXPIRATION_DATE
final ActionType ActionType

        public
        static UPDATE_OWNER
final ActionType ActionType

        public
        static UPDATE_PRIORITY
final ActionType ActionType

        public
        static UPDATE_REQUEST
final ActionType ActionType

        public

```

```

        static UPDATE_RESPONSE
final ActionType      ActionType

```

## Method Summary

```

    public
    static getActionType(Integer value)
ActionType

    public
    static getActionType(String value)
ActionType

    public
Integer getValue()

    public
String toString()

```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### ABORT

```
public static final ActionType ABORT
```

---

### ADD\_LISTENER

```
public static final ActionType ADD_LISTENER
```

---

### ASSIGN

```
public static final ActionType ASSIGN
```

---

### CLAIM

```
public static final ActionType CLAIM
```

---

### CLAIM\_EXPIRE

```
public static final ActionType CLAIM_EXPIRE
```

---

## COMPLETE

```
public static final ActionType COMPLETE
```

---

## CREATE

```
public static final ActionType CREATE
```

---

## EXPIRE

```
public static final ActionType EXPIRE
```

---

## REMOVE\_LISTENER

```
public static final ActionType REMOVE_LISTENER
```

---

## RESUME

```
public static final ActionType RESUME
```

---

## RETURN

```
public static final ActionType RETURN
```

---

## START

```
public static final ActionType START
```

---

## STOP

```
public static final ActionType STOP
```

---

## SUSPEND

```
public static final ActionType SUSPEND
```

---

## UPDATE\_CAN\_BE\_ABORTED

```
public static final ActionType UPDATE_CAN_BE_ABORTED
```

---

## UPDATE\_CAN\_BE\_REASSIGNED

```
public static final ActionType UPDATE_CAN_BE_REASSIGNED
```

---

## UPDATE\_CAN\_BE\_RETURNED

```
public static final ActionType UPDATE_CAN_BE_RETURNED
```

---

## UPDATE\_CLAIM\_DATE

```
public static final ActionType UPDATE_CLAIM_DATE
```

---

## UPDATE\_COMMENT

```
public static final ActionType UPDATE_COMMENT
```

---

## UPDATE\_EXPIRATION\_DATE

```
public static final ActionType UPDATE_EXPIRATION_DATE
```

---

## UPDATE\_OWNER

```
public static final ActionType UPDATE_OWNER
```

---

## UPDATE\_PRIORITY

```
public static final ActionType UPDATE_PRIORITY
```

---

## UPDATE\_REQUEST

```
public static final ActionType UPDATE_REQUEST
```

---

## UPDATE\_RESPONSE

```
public static final ActionType UPDATE_RESPONSE
```

## Method Detail

### getActionType(Integer) Method

```
public static ActionType getActionType(Integer value)
```

---

### getActionType(String) Method

```
public static ActionType getActionType(String value)
```

---

### getValue() Method

```
public Integer getValue()
```

---

### toString() Method

```
public String toString()
```

#### *Overrides*

```
    Object.toString()
```

# Assignee Class

*public class Assignee*

extends Object  
implements Serializable

This class provide information about the task assignee.

---

## ***Hierarchy***

Object  
Assignee

## ***All Implemented Interfaces***

Serializable

---

## **Field Summary**

protected String  
                    **groups**  
                            String  
            public  
volatile int **hashCode**  
                    int  
protected String  
                    **users**  
                            String

## **Constructor Summary**

*Assignee()*

## **Method Summary**

public  
boolean **equals**(Object other)  
public  
String[] **getAll**()



```

    public
String[] getGroups()
    public
String[] getUsers()
    public
    int hashCode()
    public
    void setGroups(String[] groups)
    public
    void setUsers(String[] users)

```

#### Methods from class `java.lang.Object`

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### groups

`protected String groups`

---

### hashCode

`public volatile int hashCode`

---

### users

`protected String users`

## Constructor Detail

### Assignee

`public Assignee()`

## Method Detail

### `equals(Object)` Method

`public boolean equals(Object other)`

***Overrides***

Object.equals(Object)

---

**getAll() Method**

```
public String[] getAll()
```

---

**getGroups() Method**

```
public String[] getGroups()
```

---

**getUsers() Method**

```
public String[] getUsers()
```

---

**hashCode() Method**

```
public int hashCode()
```

***Overrides***

Object.hashCode()

---

**setGroups(String[]) Method**

```
public void setGroups(String[] groups)
```

---

**setUsers(String[]) Method**

```
public void setUsers(String[] users)
```

# AssignmentException Class

*public class AssignmentException*

extends Exception

Exception thrown when an assignment operation fails.

---

## ***Hierarchy***

Object  
  Throwable  
    Exception  
      AssignmentException

## ***All Implemented Interfaces***

Serializable

---

## **Constructor Summary**

***AssignmentException(Exception except)***

Construct a new ProcessAssignmentException exception wrapping an underlying exception.

***AssignmentException(String message, Exception except)***

Construct a new AssignmentException exception wrapping an underlying exception and providing a message.

***AssignmentException(String message)***

Construct a new AssignmentException exception with a message.

## **Method Summary**

public  
Exception ***getException()***

Returns the underlying exception, if this exception wraps another exception.

```

public getMessage()
String      Return a detail message for this exception.
public
    void printStackTrace()
public
    void printStackTrace(PrintStream stream)
public
    void printStackTrace(PrintWriter writer)
public
String toString()
        Override toString to pick up any embedded exception.
    
```

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`,  
`initCause`, `setStackTrace`,

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### AssignmentException

```
public AssignmentException(Exception except)
```

Construct a new ProcessAssignmentException exception wrapping an underlying exception.

---

### AssignmentException

```
public AssignmentException(String message,
                           Exception except)
```

Construct a new AssignmentException exception wrapping an underlying exception and providing a message.

---

### AssignmentException

```
public AssignmentException(String message)
```

Construct a new AssignmentException exception with a message.

## Method Detail

### getException() Method

```
public Exception getException()
```

Returns the underlying exception, if this exception wraps another exception.

#### Returns

*The underlying exception, or null*

---

### getMessage() Method

```
public String getMessage()
```

Return a detail message for this exception.

If there is an embedded exception, and if the NestedException has no detail message of its own, this method will return the detail message from the embedded exception.

#### Overrides

Throwable.getMessage()

#### Returns

*The error or warning message.*

---

### printStackTrace() Method

```
public void printStackTrace()
```

#### Overrides

Throwable.printStackTrace()

---

### printStackTrace(PrintStream) Method

```
public void printStackTrace(PrintStream stream)
```

#### Overrides

Throwable.printStackTrace(PrintStream)

---

## **printStackTrace(PrintWriter) Method**

```
public void printStackTrace(PrintWriter writer)
```

### ***Overrides***

Throwable.printStackTrace(PrintWriter)

---

## **toString() Method**

```
public String toString()
```

Override toString to pick up any embedded exception.

### ***Overrides***

Throwable.toString()

### **Returns**

*A string representation of this exception.*

# BusinessTime Class

*public final class **BusinessTime***

extends Object  
implements Serializable

This class is used to store a business time. A business time is composed of a duration and a business calendar. A valid duration is for example 0min0hour0day. If no calendarName and no username are specified the system calendar will be used by the worklist to compute the effective time. If a calendarName is specified, this calendar will be used. If a user is specified, this user's calendar will be used except if the user has no calendar, in this case the system calendar will be used. You can't specify a username and a calendarId at the same time.

---

## **Hierarchy**

Object  
BusinessTime

## **All Implemented Interfaces**

Serializable

---

## Constructor Summary

***BusinessTime()***

## Method Summary

```
public
String getCalendarName()
public
String getDuration()
public
String getUserName()
public
void setCalendarName(String calendarName)
public
void setDuration(String duration)
public
void setUserName(String userName)
```

**Methods from class `java.lang.Object`**

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### BusinessTime

```
public BusinessTime()
```

## Method Detail

### getCalendarName() Method

```
public String getCalendarName()
```

---

### getDuration() Method

```
public String getDuration()
```

---

### getUserName() Method

```
public String getUserName()
```

---

### setCalendarName(String) Method

```
public void setCalendarName(String calendarName)
```

---

### setDuration(String) Method

```
public void setDuration(String duration)
```

---

### setUserName(String) Method

```
public void setUserName(String userName)
```



# ManagementException Class

*public class ManagementException*

extends Exception

Exception thrown when an operation attempted on the Worklist Manager fails.

---

## **Hierarchy**

Object  
  Throwable  
    Exception  
      ManagementException

## **All Implemented Interfaces**

Serializable

---

## **Constructor Summary**

***ManagementException(Exception except)***

Construct a new ProcessManagementException exception wrapping an underlying exception.

***ManagementException(String message, Exception except)***

Construct a new ManagementException exception wrapping an underlying exception and providing a message.

***ManagementException(String message)***

Construct a new ManagementException exception with a message.

## **Method Summary**

public  
Exception ***getException()***

Returns the underlying exception, if this exception wraps another exception.

```

public getMessage()
String      Return a detail message for this exception.
public
    void printStackTrace()
public
    void printStackTrace(PrintStream stream)
public
    void printStackTrace(PrintWriter writer)
public
String toString()
        Override toString to pick up any embedded exception.

```

#### Methods from `java.lang.Throwable`

`fillInStackTrace`, `getCause`, `getLocalizedMessage`, `getStackTrace`,  
`initCause`, `setStackTrace`,

#### Methods from class `java.lang.Object`

`clone`, `equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`,  
`toString`, `wait`, `wait`, `wait`

## Constructor Detail

### ManagementException

```
public ManagementException(Exception except)
```

Construct a new `ProcessManagementException` exception wrapping an underlying exception.

---

### ManagementException

```
public ManagementException(String message,
                           Exception except)
```

Construct a new `ManagementException` exception wrapping an underlying exception and providing a message.

---

### ManagementException

```
public ManagementException(String message)
```

Construct a new `ManagementException` exception with a message.

## Method Detail

### getException() Method

```
public Exception getException()
```

Returns the underlying exception, if this exception wraps another exception.

#### Returns

*The underlying exception, or null*

---

### getMessage() Method

```
public String getMessage()
```

Return a detail message for this exception.

If there is an embedded exception, and if the NestedException has no detail message of its own, this method will return the detail message from the embedded exception.

#### Overrides

Throwable.getMessage()

#### Returns

*The error or warning message.*

---

### printStackTrace() Method

```
public void printStackTrace()
```

#### Overrides

Throwable.printStackTrace()

---

### printStackTrace(PrintStream) Method

```
public void printStackTrace(PrintStream stream)
```

#### Overrides

Throwable.printStackTrace(PrintStream)

---

## **printStackTrace(PrintWriter) Method**

```
public void printStackTrace(PrintWriter writer)
```

### ***Overrides***

Throwable.printStackTrace(PrintWriter)

---

## **toString() Method**

```
public String toString()
```

Override toString to pick up any embedded exception.

### ***Overrides***

Throwable.toString()

### **Returns**

*A string representation of this exception.*

# StateType Class

*public class StateType*

extends Object  
implements Serializable

This class describes the different possible states of a task. There are 6 possible states. Please refer to the documentation to learn more about states and state transitions.

---

## **Hierarchy**

Object  
StateType

## **All Implemented Interfaces**

Serializable

---

## Field Summary

```
public
static ABORTED
final StateType      StateType

public
static ASSIGNED
final StateType      StateType

public
static CLAIMED
final StateType      StateType

public
static COMPLETED
final StateType      StateType

public
static STARTED
final StateType      StateType

public
static SUSPENDED
final StateType      StateType
```

## Constructor

### Summary

*StateType(String name, Integer value)*

## Method

### Summary

```

    public
    static getStateType(Integer value)
StateType
    public
    static getStateType(String value)
StateType
    public
Integer getValue()
    public
String toString()

```

#### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, wait, wait, wait

## Field Detail

### ABORTED

```
public static final StateType ABORTED
```

---

### ASSIGNED

```
public static final StateType ASSIGNED
```

---

### CLAIMED

```
public static final StateType CLAIMED
```

---

### COMPLETED

```
public static final StateType COMPLETED
```

---

## STARTED

```
public static final StateType STARTED
```

---

## SUSPENDED

```
public static final StateType SUSPENDED
```

## Constructor Detail

### StateType

```
public StateType(String name,  
                 Integer value)
```

## Method Detail

### getStateType(Integer) Method

```
public static StateType getStateType(Integer value)
```

---

### getStateType(String) Method

```
public static StateType getStateType(String value)
```

---

### getValue() Method

```
public Integer getValue()
```

---

### toString() Method

```
public String toString()
```

#### *Overrides*

```
    Object.toString()
```

# TaskCreation Class

*public final class TaskCreation*

extends Object  
implements Cloneable, Serializable

This class provide information for creatin a task. It is necessary to specify at least the task name. if not specified some task property values are picked by default: priority = 1 canBeReassigned, canBeReturned and CanBeAborted are true. Footnote when a task is created if no onwer is specified the ejb caller will be the task owner.

---

## Hierarchy

Object  
TaskCreation

## All Implemented Interfaces

Cloneable, Serializable

---

## Constructor Summary

*TaskCreation()*

## Method Summary

```
public
Object clone()
public
String getAssignToUser()
public
String getAssignToUserInGroup()
public
String[] getAssignToUsersAndGroups()
public
Boolean getCanBeAborted()
public
Boolean getCanBeReassigned()
public
Boolean getCanBeReturned()
```



```

        public
BusinessTime getClaimDueBusinessTime()
        public Date
                getClaimDueDate()
        public
        String getComment()
        public
BusinessTime getCompletionDueBusinessTime()
        public Date
                getCompletionDueDate()
        public
        String getDescription()
        public
        String getName()
        public
        String getOwner()
        public
        Integer getPriority()
        public
TaskMessage getRequest()
        public void
                setAssignToUser(String assignToUser)
        public void
                setAssignToUserInGroup(String assignToUserInGroup)
        public void
                setAssignToUsersAndGroups(String[] assignToUsersAndGroups)
        public void
                setCanBeAborted(Boolean canBeAborted)
        public void
                setCanBeReassigned(Boolean canBeReassigned)
        public void
                setCanBeReturned(Boolean canBeReturned)
        public void
                setClaimDueBusinessTime(BusinessTime claimDueBusinessTime)
        public void
                setClaimDueDate(Date claimDueDate)
        public void
                setComment(String comment)
        public void
                setCompletionDueBusinessTime(BusinessTime completionDueBusinessTime)
        public void
                setCompletionDueDate(Date completionDueDate)
        public void
                setDescription(String description)
        public void
                setName(String name)

```

```

public void setOwner(String owner)
public void
    setPriority(Integer priority)
public void
    setRequest(TaskMessage request)

```

#### Methods from class `java.lang.Object`

`equals`, `finalize`, `getClass`, `hashCode`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Constructor Detail

### TaskCreation

```
public TaskCreation()
```

## Method Detail

### `clone()` Method

```
public Object clone()
```

#### *Overrides*

```
Object.clone()
```

---

### `getAssignToUser()` Method

```
public String getAssignToUser()
```

---

### `getAssignToUserInGroup()` Method

```
public String getAssignToUserInGroup()
```

---

### `getAssignToUsersAndGroups()` Method

```
public String[] getAssignToUsersAndGroups()
```

---

### `getCanBeAborted()` Method

```
public Boolean getCanBeAborted()
```

---

**getCanBeReassigned() Method**

```
public Boolean getCanBeReassigned()
```

---

**getCanBeReturned() Method**

```
public Boolean getCanBeReturned()
```

---

**getClaimDueBusinessTime() Method**

```
public BusinessTime getClaimDueBusinessTime()
```

---

**getClaimDueDate() Method**

```
public Date getClaimDueDate()
```

---

**getComment() Method**

```
public String getComment()
```

---

**getCompletionDueBusinessTime() Method**

```
public BusinessTime getCompletionDueBusinessTime()
```

---

**getCompletionDueDate() Method**

```
public Date getCompletionDueDate()
```

---

**getDescription() Method**

```
public String getDescription()
```

---

**getName() Method**

```
public String getName()
```

---

**getOwner() Method**

```
public String getOwner()
```

---

**getPriority() Method**

```
public Integer getPriority()
```

---

**getRequest() Method**

```
public TaskMessage getRequest()
```

---

**setAssignToUser(String) Method**

```
public void setAssignToUser(String assignToUser)
```

---

**setAssignToUserInGroup(String) Method**

```
public void setAssignToUserInGroup(String assignToUserInGroup)
```

---

**setAssignToUsersAndGroups(String[]) Method**

```
public void setAssignToUsersAndGroups(String[] assignToUsersAndGroups)
```

---

**setCanBeAborted(Boolean) Method**

```
public void setCanBeAborted(Boolean canBeAborted)
```

---

**setCanBeReassigned(Boolean) Method**

```
public void setCanBeReassigned(Boolean canBeReassigned)
```

---

**setCanBeReturned(Boolean) Method**

```
public void setCanBeReturned(Boolean canBeReturned)
```

---

**setClaimDueBusinessTime(BusinessTime) Method**

```
public void setClaimDueBusinessTime(BusinessTime claimDueBusinessTime)
```

---

**setClaimDueDate(Date) Method**

```
public void setClaimDueDate(Date claimDueDate)
```

---

## setComment(String) Method

```
public void setComment(String comment)
```

---

## setCompletionDueBusinessTime(BusinessTime) Method

```
public void setCompletionDueBusinessTime(BusinessTime completionDueBusinessTime)
```

---

## setCompletionDueDate(Date) Method

```
public void setCompletionDueDate(Date completionDueDate)
```

---

## setDescription(String) Method

```
public void setDescription(String description)
```

---

## setName(String) Method

```
public void setName(String name)
```

---

## setOwner(String) Method

```
public void setOwner(String owner)
```

---

## setPriority(Integer) Method

```
public void setPriority(Integer priority)
```

---

## setRequest(TaskMessage) Method

```
public void setRequest(TaskMessage request)
```

# TaskInfo Class

*public final class TaskInfo*

extends Object  
implements Serializable

This class provide information about the runtime state of a Task. The method `getTaskInfoXMLDocument()` returns the taskInfo as an XMLObject which may be more convenient to manipulate.

---

## Hierarchy

Object  
TaskInfo

## All Implemented Interfaces

Serializable

---

## Field Summary

public String  
**owner**  
The Task  
owner

## Constructor Summary

### *TaskInfo()*

Basic constructor

*TaskInfo(String taskId, String name, String parentProcessUri, String parentProcessId, Assignee assignee, String claimant, String owner, StateType stateType, Integer priority, Date completionDueDate, Date claimDueDate, Date creationDate, Boolean canBeReassigned, Boolean canBeReturned, Boolean canBeAborted, String description, String comment, String requestType, String responseType)*

Constructor with initialization fields

## Method Summary

```

    public boolean
        equals(Object other)

    public Assignee
        getAssignee()

    public Boolean
        getCanBeAborted()

    public Boolean
        getCanBeReassigned()

    public Boolean
        getCanBeReturned()

    public String
        getClaimant()

    public Date
        getClaimDueDate()

    public String
        getComment()

    public Date
        getCompletionDueDate()

    public Date
        getCreationDate()

    public String
        getDescription()

    public String
        getName()

    public String
        getOwner()

    public String
        getParentProcessId()

    public String
        getParentProcessUri()

    public Integer
        getPriority()

    public String
        getRequestType()

    public String
        getResponseTypes()

    public StateType
        getStateType()

    public String
        getTaskId()

    public
TaskInfoXMLDocument getTaskInfoXML()

    public int

```

```

        hashCode()
public void
        setAssignee(Assignee assignee)
public void
        setCanBeAborted(Boolean canBeAborted)
public void
        setCanBeReassigned(Boolean canBeReassigned)
public void
        setCanBeReturned(Boolean canBeReturned)
public void
        setClaimant(String claimant)
public void
        setClaimDueDate(Date claimDueDate)
public void
        setComment(String comment)
public void
        setCompletionDueDate(Date completionDueDate)
public void
        setCreationDate(Date creationDate)
public void
        setDescription(String description)
public void
        setName(String name)
public void
        setOwner(String owner)
public void
        setParentProcessId(String parentProcessId)
public void
        setParentProcessUri(String parentProcessUri)
public void
        setPriority(Integer priority)
public void
        setRequestType(String requestType)
public void
        setResponseType(String responseType)
public void
        setStateType(StateType stateType)
public void
        setTaskId(String taskId)

```

#### Methods from class `java.lang.Object`

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`



## Field Detail

### owner

```
public String owner
```

The Task owner

## Constructor Detail

### TaskInfo

```
public TaskInfo()
```

Basic constructor

---

### TaskInfo

```
public TaskInfo(String taskId,
                String name,
                String parentProcessUri,
                String parentProcessId,
                Assignee assignee,
                String claimant,
                String owner,
                StateType stateType,
                Integer priority,
                Date completionDueDate,
                Date claimDueDate,
                Date creationDate,
                Boolean canBeReassigned,
                Boolean canBeReturned,
                Boolean canBeAborted,
                String description,
                String comment,
                String requestType,
                String responseType)
```

Constructor with initialization fields

## Method Detail

### equals(Object) Method

```
public boolean equals(Object other)
```

*Overrides*`Object.equals(Object)`

---

**getAssignee() Method**`public Assignee getAssignee()`

---

**getCanBeAborted() Method**`public Boolean getCanBeAborted()`

---

**getCanBeReassigned() Method**`public Boolean getCanBeReassigned()`

---

**getCanBeReturned() Method**`public Boolean getCanBeReturned()`

---

**getClaimant() Method**`public String getClaimant()`

---

**getClaimDueDate() Method**`public Date getClaimDueDate()`

---

**getComment() Method**`public String getComment()`

---

**getCompletionDueDate() Method**`public Date getCompletionDueDate()`

---

**getCreationDate() Method**`public Date getCreationDate()`

---

**getDescription() Method**

```
public String getDescription()
```

---

**getName() Method**

```
public String getName()
```

---

**getOwner() Method**

```
public String getOwner()
```

---

**getParentProcessId() Method**

```
public String getParentProcessId()
```

---

**getParentProcessUri() Method**

```
public String getParentProcessUri()
```

---

**getPriority() Method**

```
public Integer getPriority()
```

---

**getRequestType() Method**

```
public String getRequestType()
```

---

**getResponseType() Method**

```
public String getResponseType()
```

---

**getStateType() Method**

```
public StateType getStateType()
```

---

**getTaskId() Method**

```
public String getTaskId()
```

---

## getTaskInfoXML() Method

```
public TaskInfoXMLDocument getTaskInfoXML()
```

---

## hashCode() Method

```
public int hashCode()
```

### *Overrides*

```
Object.hashCode()
```

---

## setAssignee(Assignee) Method

```
public void setAssignee(Assignee assignee)
```

---

## setCanBeAborted(Boolean) Method

```
public void setCanBeAborted(Boolean canBeAborted)
```

---

## setCanBeReassigned(Boolean) Method

```
public void setCanBeReassigned(Boolean canBeReassigned)
```

---

## setCanBeReturned(Boolean) Method

```
public void setCanBeReturned(Boolean canBeReturned)
```

---

## setClaimant(String) Method

```
public void setClaimant(String claimant)
```

---

## setClaimDueDate(Date) Method

```
public void setClaimDueDate(Date claimDueDate)
```

---

## setComment(String) Method

```
public void setComment(String comment)
```

---

**setCompletionDueDate(Date) Method**

```
public void setCompletionDueDate(Date completionDueDate)
```

---

**setCreationDate(Date) Method**

```
public void setCreationDate(Date creationDate)
```

---

**setDescription(String) Method**

```
public void setDescription(String description)
```

---

**setName(String) Method**

```
public void setName(String name)
```

---

**setOwner(String) Method**

```
public void setOwner(String owner)
```

---

**setParentProcessId(String) Method**

```
public void setParentProcessId(String parentProcessId)
```

---

**setParentProcessUri(String) Method**

```
public void setParentProcessUri(String parentProcessUri)
```

---

**setPriority(Integer) Method**

```
public void setPriority(Integer priority)
```

---

**setRequestType(String) Method**

```
public void setRequestType(String requestType)
```

---

**setResponseType(String) Method**

```
public void setResponseType(String responseType)
```

---

## **setStateType(StateType) Method**

```
public void setStateType(StateType stateType)
```

---

## **setTaskId(String) Method**

```
public void setTaskId(String taskId)
```

# TaskMessage Class

*public class TaskMessage*

extends Object  
implements Serializable

This class describes a task message composed of: – a byte array that is the message itself. The mimeType qualifies the type of object that is serialized as byte[]. It is used as information to describe the message byte[] (i.e.:pdf file). The mimeType can be null.

---

## Hierarchy

Object  
TaskMessage

## All Implemented Interfaces

Serializable

---

## Field Summary

public byte  
    *message*  
        byte

public String  
    *mimeType*  
        String

## Constructor Summary

*TaskMessage(byte message, String mimeType)*

## Method Summary

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Field Detail

### message

```
public byte message
```

---

### mimeType

```
public String mimeType
```

## Constructor Detail

### TaskMessage

```
public TaskMessage(byte[] message,  
                   String mimeType)
```



# TaskSelector Class

*public final class TaskSelector*

extends Object  
implements Serializable

TaskSelector is used to select tasks based on specific properties.

A value of the interested property can be set. If not null the property will be used to filter tasks. If multiple properties are specified the task must fulfilled all the conditions to be selected.

This selector also allows to specify the number of items returned and to sort the results by property priorities.

Some special criteria accept wildcard characters: % stands for any sequence of characters \_ stands for any single character you can escape these characters using \. i.e. %a\_b\%%

---

## Hierarchy

```

Object
  TaskSelector
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Constructor Summary

### *TaskSelector()*

Constructor

***TaskSelector***(String taskIds, String taskName, boolean isNamePattern, String description, boolean isDescriptionPattern, String comment, boolean isCommentPattern, String owners, String claimants, Assignee assignee, Integer minPriority, Integer maxPriority, StateType stateTypes, String parentProcessUri, boolean isParentProcessUriPattern, String parentProcessIds, String listeningProcessUri, boolean isListeningProcessUriPattern, String listeningProcessIds, Date completionDueDateBefore, Date completionDueDateAfter, Date claimDueDateBefore, Date claimDueDateAfter, Date creationDateBefore, Date creationDateAfter, Boolean canBeReassigned, Boolean canBeReturned, Boolean canBeAborted, String propertyName, String propertyValue, boolean isPropertyValuePattern)

Constructor.

***TaskSelector**(String taskIds, String taskName, boolean isNamePattern, String description, boolean isDescriptionPattern, String comment, boolean isCommentPattern, String owners, String claimants, Assignee assignee, Integer minPriority, Integer maxPriority, StateType stateTypes, String parentProcessUri, boolean isParentProcessUriPattern, String parentProcessIds, String listeningProcessUri, boolean isListeningProcessUriPattern, String listeningProcessIds, Date completionDueDateBefore, Date completionDueDateAfter, Date claimDueDateBefore, Date claimDueDateAfter, Date creationDateBefore, Date creationDateAfter, Boolean canBeReassigned, Boolean canBeReturned, Boolean canBeAborted, String propertyNames, String propertyValues, boolean isPropertyValuePattern)*

Constructor.

## Method Summary

```

public
    void addProperties(String[] propertyNames, String[] propertyValues, boolean[]
        isPatternFlags)

public
    void addProperty(String propertyName, String propertyValue, boolean isPattern)

public
Assignee getAssignee()

public
Boolean getCanBeAborted()

public
Boolean getCanBeReassigned()

public
Boolean getCanBeReturned()

public
String[] getClaimants()

public
    Date getClaimDueDateAfter()

public
    Date getClaimDueDateBefore()

public
String getComment()

public
    Date getCompletionDueDateAfter()

public
    Date getCompletionDueDateBefore()

public
    Date getCreationDateAfter()

public
    Date getCreationDateBefore()

public
String getDescription()

```

```

    public
String[] getListeningProcessIds()
    public
String getListeningProcessUri()
    public
Integer getMaxPriority()
    public
    int getMaxTasksReturned()
    public
Integer getMinPriority()
    public
String[] getOwners()
    public
String[] getParentProcessIds()
    public
String getParentProcessUri()
    public
String getPropertyName()
    public
String[] getPropertyNames()
    public
String getPropertyValue()
    public
String getPropertyValue(String propertyName)
        throws IllegalArgumentException if the property name has not previously been set.
    public
    short getSortByAssignee()
    public
    short getSortByCanBeAborted()
    public
    short getSortByCanBeReassigned()
    public
    short getSortByCanBeReturned()
    public
    short getSortByClaimant()
    public
    short getSortByClaimDueDate()
    public
    short getSortByComment()
    public
    short getSortByCompletionDueDate()
    public
    short getSortByCreationDate()
    public
    short getSortByDescription()

```

```

public getSortByListeningProcessId()
    short
public
    short getSortByListeningProcessUri()
public
    short getSortByOwner()
public
    short getSortByParentProcessId()
public
    short getSortByParentProcessUri()
public
    short getSortByPriority()
public
    short getSortByPropertyValue()
public
    short getSortByStateType()
public
    short getSortByTaskId()
public
    short getSortByTaskName()
public
StateType[] getStateTypes()
public
String[] getTaskIds()
public
String getTaskName()
public
boolean isCommentPattern()
public
boolean isDescriptionPattern()
public
boolean isListeningProcessUriPattern()
public
boolean isParentProcessUriPattern()
public
boolean isPropertyValuePattern()
    previously been set.
public
boolean isPropertyValuePattern(String propertyName)
    throws IllegalArgumentException if the property name has not previously been set.
public
boolean isTaskNamePattern()
public
    void setAssignee(Assignee assignee)
public
    void setCanBeAborted(Boolean canBeAborted)

```

```

public
    void setCanBeReassigned(Boolean canBeReassigned)
public
    void setCanBeReturned(Boolean canBeReturned)
public
    void setClaimants(String[] claimants)
public
    void setClaimDueDateAfter(Date claimDueDateAfter)
public
    void setClaimDueDateBefore(Date claimDueDateBefore)
public
    void setComment(String comment, boolean isPattern)
public
    void setCompletionDueDateAfter(Date completionDueDateAfter)
public
    void setCompletionDueDateBefore(Date completionDueDateBefore)
public
    void setCreationDateAfter(Date creationDateAfter)
public
    void setCreationDateBefore(Date creationDateBefore)
public
    void setDescription(String description, boolean isPattern)
public
    void setListeningProcessIds(String[] listeningProcessIds)
public
    void setListeningProcessUri(String listeningProcessUri, boolean isPattern)
public
    void setMaxPriority(Integer maxPriority)
public
    void setMaxTasksReturned(int count)
public
    void setMinPriority(Integer minPriority)
public
    void setOwners(String[] owners)
public
    void setParentProcessIds(String[] parentProcessIds)
public
    void setParentProcessUri(String parentProcessUri, boolean isPattern)
public
    void setPropertyName(String propertyName)
public
    void setPropertyNames(String[] propertyNames)
public
    void setPropertyValue(String propertyName, String propertyValue, boolean isPattern)
        throws IllegalArgumentException if no property name has previously been set.

```

## Java Class Reference

```
public setPropertyValue(String propertyValue, boolean isPattern)
    void          throws IllegalArgumentException if the property name has not previously been set.
public
    void setSortByAssignee(short order)
public
    void setSortByCanBeAborted(short order)
public
    void setSortByCanBeReassigned(short order)
public
    void setSortByCanBeReturned(short order)
public
    void setSortByClaimant(short order)
public
    void setSortByClaimDueDate(short order)
public
    void setSortByComment(short order)
public
    void setSortByCompletionDueDate(short order)
public
    void setSortByCreationDate(short order)
public
    void setSortByDescription(short order)
public
    void setSortByListeningProcessId(short order)
public
    void setSortByListeningProcessUri(short order)
public
    void setSortByOwner(short order)
public
    void setSortByParentProcessId(short order)
public
    void setSortByParentProcessUri(short order)
public
    void setSortByPriority(short order)
public
    void setSortByPropertyValue(short order)
public
    void setSortByStateType(short order)
public
    void setSortByTaskId(short order)
public
    void setSortByTaskName(short order)
public
    void setStateTypes(StateType[] stateTypes)
public
    void setTaskIds(String[] taskIds)
```

```
public
void setTaskName(String taskName, boolean isPattern)
```

### Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll, toString, wait, wait, wait

## Constructor Detail

### TaskSelector

```
public TaskSelector()
```

Constructor

---

### TaskSelector

```
public TaskSelector(String[] taskIds,
                    String taskName,
                    boolean isNamePattern,
                    String description,
                    boolean isDescriptionPattern,
                    String comment,
                    boolean isCommentPattern,
                    String[] owners,
                    String[] claimants,
                    Assignee assignee,
                    Integer minPriority,
                    Integer maxPriority,
                    StateType[] stateTypes,
                    String parentProcessUri,
                    boolean isParentProcessUriPattern,
                    String[] parentProcessIds,
                    String listeningProcessUri,
                    boolean isListeningProcessUriPattern,
                    String[] listeningProcessIds,
                    Date completionDueDateBefore,
                    Date completionDueDateAfter,
                    Date claimDueDateBefore,
                    Date claimDueDateAfter,
                    Date creationDateBefore,
                    Date creationDateAfter,
                    Boolean canBeReassigned,
                    Boolean canBeReturned,
                    Boolean canBeAborted,
                    String propertyName,
                    String propertyValue,
                    boolean isPropertyValuePattern)
```

Constructor. The null parameters are not used to filter the Tasks.

---

## TaskSelector

```
public TaskSelector(String[] taskIds,
    String taskName,
    boolean isNamePattern,
    String description,
    boolean isDescriptionPattern,
    String comment,
    boolean isCommentPattern,
    String[] owners,
    String[] claimants,
    Assignee assignee,
    Integer minPriority,
    Integer maxPriority,
    StateType[] stateTypes,
    String parentProcessUri,
    boolean isParentProcessUriPattern,
    String[] parentProcessIds,
    String listeningProcessUri,
    boolean isListeningProcessUriPattern,
    String[] listeningProcessIds,
    Date completionDueDateBefore,
    Date completionDueDateAfter,
    Date claimDueDateBefore,
    Date claimDueDateAfter,
    Date creationDateBefore,
    Date creationDateAfter,
    Boolean canBeReassigned,
    Boolean canBeReturned,
    Boolean canBeAborted,
    String[] propertyNames,
    String[] propertyValues,
    boolean[] isPropertyValuePattern)
```

Constructor. The null parameters are not used to filter the Tasks.

## Method Detail

### addProperties(String[], String[], boolean[]) Method

```
public void addProperties(String[] propertyNames,
    String[] propertyValues,
    boolean[] isPatternFlags)
```

---

### addProperty(String, String, boolean) Method

```
public void addProperty(String propertyName,
    String propertyValue,
    boolean isPattern)
```

---



**getAssignee() Method**

```
public Assignee getAssignee()
```

---

**getCanBeAborted() Method**

```
public Boolean getCanBeAborted()
```

---

**getCanBeReassigned() Method**

```
public Boolean getCanBeReassigned()
```

---

**getCanBeReturned() Method**

```
public Boolean getCanBeReturned()
```

---

**getClaimants() Method**

```
public String[] getClaimants()
```

---

**getClaimDueDateAfter() Method**

```
public Date getClaimDueDateAfter()
```

---

**getClaimDueDateBefore() Method**

```
public Date getClaimDueDateBefore()
```

---

**getComment() Method**

```
public String getComment()
```

---

**getCompletionDueDateAfter() Method**

```
public Date getCompletionDueDateAfter()
```

---

**getCompletionDueDateBefore() Method**

```
public Date getCompletionDueDateBefore()
```

---

**getCreationDateAfter() Method**

```
public Date getCreationDateAfter()
```

---

**getCreationDateBefore() Method**

```
public Date getCreationDateBefore()
```

---

**getDescription() Method**

```
public String getDescription()
```

---

**getListeningProcessIds() Method**

```
public String[] getListeningProcessIds()
```

---

**getListeningProcessUri() Method**

```
public String getListeningProcessUri()
```

---

**getMaxPriority() Method**

```
public Integer getMaxPriority()
```

---

**getMaxTasksReturned() Method**

```
public int getMaxTasksReturned()
```

---

**getMinPriority() Method**

```
public Integer getMinPriority()
```

---

**getOwners() Method**

```
public String[] getOwners()
```

---

**getParentProcessIds() Method**

```
public String[] getParentProcessIds()
```

---

## getParentProcessUri() Method

```
public String getParentProcessUri()
```

---

## getPropertyName() Method

**DEPRECATED** Use `getPropertyNames` instead.

```
public String getPropertyName()
```

### Returns

*A single property name.*

### Related Topics

`getPropertyNames`

---

## getPropertyNames() Method

```
public String[] getPropertyNames()
```

### Returns

*An array of property names that have been set or null if none have been set.*

---

## getPropertyValue() Method

**DEPRECATED** Use `getPropertyValue(String propertyName)`

```
public String getPropertyValue()
```

### Returns

*property value*

---

## getPropertyValue(String) Method

**DEPRECATED** Use `setPropertyValue(String propertyName, String propertyValue, boolean isPattern)`

```
public String getPropertyValue(String propertyName)
```

throws `IllegalArgumentException` if the property name has not previously been set.

---

**getSortByAssignee() Method**

```
public short getSortByAssignee()
```

---

**getSortByCanBeAborted() Method**

```
public short getSortByCanBeAborted()
```

---

**getSortByCanBeReassigned() Method**

```
public short getSortByCanBeReassigned()
```

---

**getSortByCanBeReturned() Method**

```
public short getSortByCanBeReturned()
```

---

**getSortByClaimant() Method**

```
public short getSortByClaimant()
```

---

**getSortByClaimDueDate() Method**

```
public short getSortByClaimDueDate()
```

---

**getSortByComment() Method**

```
public short getSortByComment()
```

---

**getSortByCompletionDueDate() Method**

```
public short getSortByCompletionDueDate()
```

---

**getSortByCreationDate() Method**

```
public short getSortByCreationDate()
```

---

**getSortByDescription() Method**

```
public short getSortByDescription()
```

---

**getSortByListeningProcessId() Method**

```
public short getSortByListeningProcessId()
```

---

**getSortByListeningProcessUri() Method**

```
public short getSortByListeningProcessUri()
```

---

**getSortByOwner() Method**

```
public short getSortByOwner()
```

---

**getSortByParentProcessId() Method**

```
public short getSortByParentProcessId()
```

---

**getSortByParentProcessUri() Method**

```
public short getSortByParentProcessUri()
```

---

**getSortByPriority() Method**

```
public short getSortByPriority()
```

---

**getSortByPropertyValue() Method**

```
public short getSortByPropertyValue()
```

---

**getSortByStateType() Method**

```
public short getSortByStateType()
```

---

**getSortByTaskId() Method**

```
public short getSortByTaskId()
```

---

**getSortByTaskName() Method**

```
public short getSortByTaskName()
```

---

## getStateTypes() Method

```
public StateType[] getStateTypes()
```

---

## getTaskIds() Method

```
public String[] getTaskIds()
```

---

## getTaskName() Method

```
public String getTaskName()
```

---

## isCommentPattern() Method

```
public boolean isCommentPattern()
```

---

## isDescriptionPattern() Method

```
public boolean isDescriptionPattern()
```

---

## isListeningProcessUriPattern() Method

```
public boolean isListeningProcessUriPattern()
```

---

## isParentProcessUriPattern() Method

```
public boolean isParentProcessUriPattern()
```

---

## isPropertyValuePattern() Method

**DEPRECATED** Use `isPropertyValuePattern(String propertyName)`

```
public boolean isPropertyValuePattern()
```

previously been set.

---

## isPropertyValuePattern(String) Method

**DEPRECATED** Use `isPropertyValuePattern(String propertyName)`

```
public boolean isPropertyValuePattern(String propertyName)
```

throws `IllegalArgumentException` if the property name has not previously been set.

---

### **isTaskNamePattern() Method**

```
public boolean isTaskNamePattern()
```

---

### **setAssignee(Assignee) Method**

```
public void setAssignee(Assignee assignee)
```

---

### **setCanBeAborted(Boolean) Method**

```
public void setCanBeAborted(Boolean canBeAborted)
```

---

### **setCanBeReassigned(Boolean) Method**

```
public void setCanBeReassigned(Boolean canBeReassigned)
```

---

### **setCanBeReturned(Boolean) Method**

```
public void setCanBeReturned(Boolean canBeReturned)
```

---

### **setClaimants(String[]) Method**

```
public void setClaimants(String[] claimants)
```

---

### **setClaimDueDateAfter(Date) Method**

```
public void setClaimDueDateAfter(Date claimDueDateAfter)
```

---

### **setClaimDueDateBefore(Date) Method**

```
public void setClaimDueDateBefore(Date claimDueDateBefore)
```

---

### **setComment(String, boolean) Method**

```
public void setComment(String comment,  
                       boolean isPattern)
```

---

## setCompletionDueDateAfter(Date) Method

```
public void setCompletionDueDateAfter(Date completionDueDateAfter)
```

---

## setCompletionDueDateBefore(Date) Method

```
public void setCompletionDueDateBefore(Date completionDueDateBefore)
```

---

## setCreationDateAfter(Date) Method

```
public void setCreationDateAfter(Date creationDateAfter)
```

---

## setCreationDateBefore(Date) Method

```
public void setCreationDateBefore(Date creationDateBefore)
```

---

## setDescription(String, boolean) Method

```
public void setDescription(String description,  
                           boolean isPattern)
```

---

## setListeningProcessIds(String[]) Method

```
public void setListeningProcessIds(String[] listeningProcessIds)
```

---

## setListeningProcessUri(String, boolean) Method

```
public void setListeningProcessUri(String listeningProcessUri,  
                                   boolean isPattern)
```

---

## setMaxPriority(Integer) Method

```
public void setMaxPriority(Integer maxPriority)
```

---

## setMaxTasksReturned(int) Method

```
public void setMaxTasksReturned(int count)
```

---



## setMinPriority(Integer) Method

```
public void setMinPriority(Integer minPriority)
```

---

## setOwners(String[]) Method

```
public void setOwners(String[] owners)
```

---

## setParentProcessIds(String[]) Method

```
public void setParentProcessIds(String[] parentProcessIds)
```

---

## setParentProcessUri(String, boolean) Method

```
public void setParentProcessUri(String parentProcessUri,  
                                boolean isPattern)
```

---

## setPropertyName(String) Method

```
public void setPropertyName(String propertyName)
```

---

## setPropertyNames(String[]) Method

```
public void setPropertyNames(String[] propertyName)
```

---

## setPropertyValue(String, String, boolean) Method

```
public void setPropertyValue(String propertyName,  
                             String propertyValue,  
                             boolean isPattern)
```

throws `IllegalArgumentException` if no property name has previously been set.

---

## setPropertyValue(String, boolean) Method

**DEPRECATED** Use `setPropertyValue(String propertyName, String propertyValue, boolean isPattern)`

```
public void setPropertyValue(String propertyValue,  
                             boolean isPattern)
```

throws `IllegalArgumentException` if the property name has not previously been set.

---

**setSortByAssignee(short) Method**

```
public void setSortByAssignee(short order)
```

---

**setSortByCanBeAborted(short) Method**

```
public void setSortByCanBeAborted(short order)
```

---

**setSortByCanBeReassigned(short) Method**

```
public void setSortByCanBeReassigned(short order)
```

---

**setSortByCanBeReturned(short) Method**

```
public void setSortByCanBeReturned(short order)
```

---

**setSortByClaimant(short) Method**

```
public void setSortByClaimant(short order)
```

---

**setSortByClaimDueDate(short) Method**

```
public void setSortByClaimDueDate(short order)
```

---

**setSortByComment(short) Method**

```
public void setSortByComment(short order)
```

---

**setSortByCompletionDueDate(short) Method**

```
public void setSortByCompletionDueDate(short order)
```

---

**setSortByCreationDate(short) Method**

```
public void setSortByCreationDate(short order)
```

---

**setSortByDescription(short) Method**

```
public void setSortByDescription(short order)
```

---

**setSortByListeningProcessId(short) Method**

```
public void setSortByListeningProcessId(short order)
```

---

**setSortByListeningProcessUri(short) Method**

```
public void setSortByListeningProcessUri(short order)
```

---

**setSortByOwner(short) Method**

```
public void setSortByOwner(short order)
```

---

**setSortByParentProcessId(short) Method**

```
public void setSortByParentProcessId(short order)
```

---

**setSortByParentProcessUri(short) Method**

```
public void setSortByParentProcessUri(short order)
```

---

**setSortByPriority(short) Method**

```
public void setSortByPriority(short order)
```

---

**setSortByPropertyValue(short) Method**

```
public void setSortByPropertyValue(short order)
```

---

**setSortByStateType(short) Method**

```
public void setSortByStateType(short order)
```

---

**setSortByTaskId(short) Method**

```
public void setSortByTaskId(short order)
```

---

**setSortByTaskName(short) Method**

```
public void setSortByTaskName(short order)
```

---

## **setStateTypes(StateType[]) Method**

```
public void setStateTypes(StateType[] stateTypes)
```

---

## **setTaskIds(String[]) Method**

```
public void setTaskIds(String[] taskIds)
```

---

## **setTaskName(String, boolean) Method**

```
public void setTaskName(String taskName,  
                        boolean isPattern)
```

# WorklistManager Interface

*public interface WorklistManager*

## Method Summary

```

public
    void abortTask(String taskId)
        Abort a task

public
    void archiveTasks()
        Archive all the tasks in their final state (COMPLETED or ABORTED) if the
        archiver is on.

public
    void assignToUser(String userName, String taskId)
        Assign the task to the user and claim it on his behalf.

public
    String assignToUserInGroup(String groupName, String taskId)
        Assign the task to the selected user in the group and claim the task on his behalf.

public
    void assignToUsersAndGroups(String[] userNamesAndOrGroupNames, String
        taskId)
        Assign the tasks to users and/or groups

public
    void claimTask(String taskId)
        Claim the task.

public
    void claimTask(String user, String taskId)
        Claim the task on the behalf on the user for the worklist admin or the task owner.

public
    void completeTask(String taskId)
        Complete a task in the state STARTED.

public
    String createTask(String name)
        Create a task, with all the default values.

public
    String createTask(TaskCreation taskCreation)
        Create a task.

public
    String createTaskXML(TaskCreationXMLDocument doc)
        Create task using an xml document

public
    void deleteTask(String taskId)

```

## Java Class Reference

Force the deletion of a task.

```
public
    void deleteTasks(TaskSelector selector)
        Force the deletion of the task selected.
```

```
public
String[] getContainingGroups(String userOrGroup)
    Util method.
```

```
public int
    getSelectedTasksNumber(TaskSelector selector)
        Get the number of tasks selected.
```

```
public
String[] getTaskIds(TaskSelector selector)
    Get a String array of taskIds of all the selected tasks
```

```
public
TaskInfo getTaskInfo(String taskId)
    Get the task info object.
```

```
public
TaskInfo[] getTaskInfos(TaskSelector selector)
    Get a task info array of all the selected tasks
```

```
public
String[] getTaskProperties(String[] propertyNames, String taskId)
    Get user task property values.
```

```
public
String[] getTaskPropertiesNames(String taskId)
    Return all the user task property names already created.
```

```
public
String getTaskProperty(String propertyName, String taskId)
    Get a user task property value.
```

```
public
TaskMessage getTaskRequest(String taskId)
    Get the task request.
```

```
public
XmlObject getTaskRequestAsXmlObject(String taskId)
    Get the task request as XmlObject
```

```
public
TaskMessage getTaskResponse(String taskId)
    Get the task response.
```

```
public
XmlObject getTaskResponseAsXmlObject(String taskId)
    Get the task response as XmlObject.
```

```
public
String[] getUsersInGroup(String group)
    Util method.
```

```
public
void purgeTasks()
    Purge all the tasks that have been in their final state for more than the delayToPurge value.
```

## Java Class Reference

```
public removeTaskProperties(String[] propertyNames, String taskId)
String[]      Remove user task properties

public
void resumeTask(String taskId)
              Return a task in the state it was before to have been suspended.

public
void returnTask(String taskId)
              Return a task that is in the state CLAIMED or STARTED to the state ASSIGNED
              with all the previous assignees.

public
void setTaskCanBeAborted(Boolean value, String taskId)
              Specify if an assignee or the claimant can abort the task.

public
void setTaskCanBeReassigned(Boolean value, String taskId)
              Specify if a task assignee can reassign the task.

public
void setTaskCanBeReturned(Boolean value, String taskId)
              Specify if a the claimant can return the task.

public
void setTaskClaimDueBusinessDate(BusinessTime time, String taskId)
              Set the task completion due date using a business time.

public
void setTaskClaimDueDate(Date claimDueDate, String taskId)
              Set the task claim due date.

public
void setTaskComment(String comment, String taskId)
              Set a task comment.

public
void setTaskCompletionDueBusinessDate(BusinessTime time, String taskId)
              Set the task completion due date using a business time.

public
void setTaskCompletionDueDate(Date completeDueDate, String taskId)
              Set the task completion due date.

public
void setTaskOwner(String owner, String taskId)
              Set the task owner.

public
void setTaskPriority(Integer priority, String taskId)
              Set a task priority

public
String[] setTaskProperties(String[] propertyNames, String[] values, String taskId)
              Update/Create user task properties.

public
String setTaskProperty(String propertyName, String value, String taskId)
              Update/Create a user task property.

public
void setTaskRequest(TaskMessage request, String taskId)
```

Set the task request.

```
public
void setTaskRequestAsXmlObject(XmlObject request, String mimeType, String
taskId)
```

Set the task request as XmlObject.

```
public
void setTaskResponse(TaskMessage response, String taskId)
```

Set the task response.

```
public
void setTaskResponseAsXmlObject(XmlObject response, String mimeType,
String taskId)
```

Set the task response as XmlObject.

```
public
void startTask(String taskId)
```

Start working on a task.

```
public
void stopTask(String taskId)
```

Return this task state from STARTED to CLAIMED.

```
public
void suspendTask(String taskId)
```

Suspend the task.

## Method Detail

### abortTask(String) Method

```
public void abortTask(String taskId)
    throws ManagementException, RemoteException
```

Abort a task

#### Exceptions

*ManagementException*  
*RemoteException*

---

### archiveTasks() Method

```
public void archiveTasks()
    throws ManagementException, RemoteException
```

Archive all the tasks in their final state (COMPLETED or ABORTED) if the archiver is on.

#### Exceptions

*ManagementException*  
*RemoteException*

---



## assignToUser(String, String) Method

```
public void assignToUser(String userName ,  
                        String taskId)  
    throws ManagementException, AssignmentException, RemoteException
```

Assign the task to the user and claim it on his behalf.

### Exceptions

*ManagementException*  
*AssignmentException*  
*RemoteException*

---

## assignToUserInGroup(String, String) Method

```
public String assignToUserInGroup(String groupName ,  
                                String taskId)  
    throws ManagementException, AssignmentException, RemoteException
```

Assign the task to the selected user in the group and claim the task on his behalf. The user who has the less task assigned is selected.

### Returns

*the user selected*

### Exceptions

*ManagementException*  
*AssignmentException*  
*RemoteException*

---

## assignToUsersAndGroups(String[], String) Method

```
public void assignToUsersAndGroups(String[] userNamesAndOrGroupNames ,  
                                String taskId)  
    throws ManagementException, AssignmentException, RemoteException
```

Assign the tasks to users and/or groups

### Exceptions

*ManagementException*  
*AssignmentException*  
*RemoteException*

---

## claimTask(String) Method

```
public void claimTask(String taskId)
    throws ManagementException, RemoteException
```

Claim the task.

### Parameters

*taskId*  
the taskId

### Exceptions

*ManagementException*  
\* @throws *RemoteException*  
*RemoteException*

---

## claimTask(String, String) Method

```
public void claimTask(String user,
    String taskId)
    throws ManagementException, RemoteException
```

Claim the task on the behalf on the user for the worklist admin or the task owner. The task will go from the state ASSIGNED to CLAIMED.

### Exceptions

*ManagementException*  
*RemoteException*

---

## completeTask(String) Method

```
public void completeTask(String taskId)
    throws ManagementException, RemoteException
```

Complete a task in the state STARTED.

### Exceptions

*ManagementException*  
*RemoteException*

---

## createTask(String) Method

```
public String createTask(String name)
    throws ManagementException, RemoteException
```

Create a task, with all the default values.

### Exceptions

*ManagementException*

If the operation fails.

*RemoteException*

---

## createTask(TaskCreation) Method

```
public String createTask(TaskCreation taskCreation)
    throws ManagementException, RemoteException, BusinessCalendarException
```

Create a task.

### Exceptions

*ManagementException*

If the operation fails.

*RemoteException*

*BusinessCalendarException*

---

## createTaskXML(TaskCreationXMLDocument) Method

```
public String createTaskXML(TaskCreationXMLDocument doc)
    throws ManagementException, BusinessCalendarException, RemoteException
```

Create task using an xml document

### Exceptions

*ManagementException*

*BusinessCalendarException*

*RemoteException*

---

## deleteTask(String) Method

```
public void deleteTask(String taskId)
    throws ManagementException, RemoteException
```

Force the deletion of a task.

### Exceptions

*ManagementException*

*RemoteException*

---

## deleteTasks(TaskSelector) Method

```
public void deleteTasks(TaskSelector selector)
    throws ManagementException, RemoteException
```

Force the deletion of the task selected.

### Exceptions

*ManagementException*  
*RemoteException*

---

## getContainingGroups(String) Method

```
public String[] getContainingGroups(String userOrGroup)
    throws ManagementException, RemoteException
```

Util method. Return all the groups containing this user or group

### Exceptions

*ManagementException*  
*RemoteException*

---

## getSelectedTasksNumber(TaskSelector) Method

```
public int getSelectedTasksNumber(TaskSelector selector)
    throws ManagementException, RemoteException
```

Get the number of tasks selected.

### Returns

*the number of task selected*

### Exceptions

*ManagementException*  
*RemoteException*

---

## getTaskIds(TaskSelector) Method

```
public String[] getTaskIds(TaskSelector selector)
    throws ManagementException, RemoteException
```

Get a String array of taskIds of all the selected tasks

## Exceptions

*ManagementException*

*RemoteException*

---

## getTaskInfo(String) Method

```
public TaskInfo getTaskInfo(String taskId)
    throws ManagementException, RemoteException
```

Get the task info object.

## Exceptions

*ManagementException*

*RemoteException*

---

## getTaskInfos(TaskSelector) Method

```
public TaskInfo[] getTaskInfos(TaskSelector selector)
    throws ManagementException, RemoteException
```

Get a task info array of all the selected tasks

## Exceptions

*ManagementException*

*RemoteException*

---

## getTaskProperties(String[], String) Method

```
public String[] getTaskProperties(String[] propertyNames,
    String taskId)
    throws ManagementException, RemoteException
```

Get user task property values.

## Exceptions

*ManagementException*

*RemoteException*

---

## getTaskPropertiesNames(String) Method

```
public String[] getTaskPropertiesNames(String taskId)
    throws ManagementException, RemoteException
```

Return all the user task property names already created.

### Exceptions

*ManagementException*

*RemoteException*

---

## getTaskProperty(String, String) Method

```
public String getTaskProperty(String propertyName,  
                             String taskId)  
    throws ManagementException, RemoteException
```

Get a user task property value. A user task property is an extension of the worklist that allows to define your own properties if the properties specified by default are not sufficient. A property is composed of 2 Strings: a name and a value. You can use the task selector to filter tasks based on property names and property values.

### Exceptions

*ManagementException*

*RemoteException*

---

## getTaskRequest(String) Method

```
public TaskMessage getTaskRequest(String taskId)  
    throws ManagementException, RemoteException
```

Get the task request.

### Exceptions

*ManagementException*

*RemoteException*

---

## getTaskRequestAsXmlObject(String) Method

```
public XmlObject getTaskRequestAsXmlObject(String taskId)  
    throws ManagementException, RemoteException
```

Get the task request as XmlObject

### Exceptions

*ManagementException*

*RemoteException*

---

## getTaskResponse(String) Method

```
public TaskMessage getTaskResponse(String taskId)
    throws ManagementException, RemoteException
```

Get the task response.

### Exceptions

*ManagementException*  
*RemoteException*

---

## getTaskResponseAsXmlObject(String) Method

```
public XmlObject getTaskResponseAsXmlObject(String taskId)
    throws ManagementException, RemoteException
```

Get the task response as XmlObject.

### Exceptions

*ManagementException*  
*RemoteException*

---

## getUsersInGroup(String) Method

```
public String[] getUsersInGroup(String group)
    throws ManagementException, RemoteException
```

Util method. Return all the members of a WLS group

### Exceptions

*ManagementException*  
*RemoteException*

---

## purgeTasks() Method

```
public void purgeTasks()
    throws ManagementException, RemoteException
```

Purge all the tasks that have been in their final state for more than the delayToPurge value. If the archiver is on and the task has not been archived yet, the task cannot be purged.

### Exceptions

*ManagementException*  
*RemoteException*

## removeTaskProperties(String[], String) Method

```
public String[] removeTaskProperties(String[] propertyNames,  
                                   String taskId)  
    throws ManagementException, RemoteException
```

Remove user task properties

### Exceptions

*ManagementException*  
*RemoteException*

---

## resumeTask(String) Method

```
public void resumeTask(String taskId)  
    throws ManagementException, RemoteException
```

Return a task in the state it was before to have been suspended.

### Exceptions

*ManagementException*  
*RemoteException*

---

## returnTask(String) Method

```
public void returnTask(String taskId)  
    throws ManagementException, RemoteException
```

Return a task that is in the state CLAIMED or STARTED to the state ASSIGNED with all the previous assignees.

### Exceptions

*ManagementException*  
*RemoteException*

---

## setTaskCanBeAborted(Boolean, String) Method

```
public void setTaskCanBeAborted(Boolean value,  
                                String taskId)  
    throws ManagementException, RemoteException
```

Specify if an assignee or the claimant can abort the task.



## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskCanBeReassigned(Boolean, String) Method

```
public void setTaskCanBeReassigned(Boolean value,  
                                   String taskId)  
    throws ManagementException, RemoteException
```

Specify if a task assignee can reassign the task.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskCanBeReturned(Boolean, String) Method

```
public void setTaskCanBeReturned(Boolean value,  
                                   String taskId)  
    throws ManagementException, RemoteException
```

Specify if a the claimant can return the task.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskClaimDueBusinessDate(BusinessTime, String) Method

```
public void setTaskClaimDueBusinessDate(BusinessTime time,  
                                         String taskId)  
    throws ManagementException, BusinessCalendarException, RemoteException
```

Set the task completion due date using a business time.

## Exceptions

*ManagementException*

*BusinessCalendarException*

*RemoteException*

---

## setTaskClaimDueDate(Date, String) Method

```
public void setTaskClaimDueDate(Date claimDueDate,  
                                String taskId)  
    throws ManagementException, RemoteException
```

Set the task claim due date.

### Exceptions

*ManagementException*

*RemoteException*

---

## setTaskComment(String, String) Method

```
public void setTaskComment(String comment,  
                            String taskId)  
    throws ManagementException, RemoteException
```

Set a task comment.

### Exceptions

*ManagementException*

*RemoteException*

---

## setTaskCompletionDueBusinessDate(BusinessTime, String) Method

```
public void setTaskCompletionDueBusinessDate(BusinessTime time,  
                                              String taskId)  
    throws ManagementException, BusinessCalendarException, RemoteException
```

Set the task completion due date using a business time.

### Exceptions

*ManagementException*

*BusinessCalendarException*

*RemoteException*

---

## setTaskCompletionDueDate(Date, String) Method

```
public void setTaskCompletionDueDate(Date completeDueDate,  
                                     String taskId)  
    throws ManagementException, RemoteException
```

Set the task completion due date.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskOwner(String, String) Method

```
public void setTaskOwner(String owner,  
                        String taskId)  
    throws ManagementException, RemoteException
```

Set the task owner. Must be worklist admin or task owner to change the owner.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskPriority(Integer, String) Method

```
public void setTaskPriority(Integer priority,  
                          String taskId)  
    throws ManagementException, RemoteException
```

Set a task priority

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskProperties(String[], String[], String) Method

```
public String[] setTaskProperties(String[] propertyNames,  
                                String[] values,  
                                String taskId)  
    throws ManagementException, RemoteException
```

Update/Create user task properties.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskProperty(String, String, String) Method

```
public String setTaskProperty(String propertyName,  
                             String value,  
                             String taskId)  
    throws ManagementException, RemoteException
```

Update/Create a user task property.

### Exceptions

*ManagementException*  
*RemoteException*

---

## setTaskRequest(TaskMessage, String) Method

```
public void setTaskRequest(TaskMessage request,  
                           String taskId)  
    throws ManagementException, RemoteException
```

Set the task request.

### Exceptions

*ManagementException*  
*RemoteException*

---

## setTaskRequestAsXmlObject(XmlObject, String, String) Method

```
public void setTaskRequestAsXmlObject(XmlObject request,  
                                      String mimeType,  
                                      String taskId)  
    throws ManagementException, RemoteException
```

Set the task request as XmlObject.

### Exceptions

*ManagementException*  
*RemoteException*

---

## setTaskResponse(TaskMessage, String) Method

```
public void setTaskResponse(TaskMessage response,  
                           String taskId)  
    throws ManagementException, RemoteException
```

Set the task response.

## Exceptions

*ManagementException*

*RemoteException*

---

## setTaskResponseAsXmlObject(XmlObject, String, String) Method

```
public void setTaskResponseAsXmlObject(XmlObject response,  
                                       String mimeType,  
                                       String taskId)  
    throws ManagementException, RemoteException
```

Set the task response as XmlObject.

## Exceptions

*ManagementException*

*RemoteException*

---

## startTask(String) Method

```
public void startTask(String taskId)  
    throws ManagementException, RemoteException
```

Start working on a task. The task must be in the state CLAIMED and will go to the state STARTED.

## Exceptions

*ManagementException*

*RemoteException*

---

## stopTask(String) Method

```
public void stopTask(String taskId)  
    throws ManagementException, RemoteException
```

Return this task state from STARTED to CLAIMED.

## Exceptions

*ManagementException*

*RemoteException*

---

## suspendTask(String) Method

```
public void suspendTask(String taskId)  
    throws ManagementException, RemoteException
```

Suspend the task. No operation are allowed on the task **SUSPENDED** until it is resumed.

### Exceptions

*ManagementException*

*RemoteException*

# WorkSubstituteManager Interface

*public interface WorkSubstituteManager*

The work substitue allows to manage the work substitute rules.

## Method Summary

```
public String[]
    addWorkSubstituteRules(WorkSubstituteRule[]
        workSubstituteRules)

    public
WorkSubstituteRule[] getAllWorkSubstituteRules()
    public String
        getWorkSubstitute(String user, Date date)
    public
WorkSubstituteRule getWorkSubstituteRule(String ruleId)
    public
WorkSubstituteRule[] getWorkSubstituteRulesWithSource(String source)
    public
WorkSubstituteRule[] getWorkSubstituteRulesWithTarget(String target)
    public void
        removeWorkSubstituteRules(String[] workSubstituteRuleIds)
    public void
        updateWorkSubstituteRules(WorkSubstituteRule[] rules)
```

## Method Detail

### **addWorkSubstituteRules(WorkSubstituteRule[]) Method**

```
public String[] addWorkSubstituteRules(WorkSubstituteRule[] workSubstituteRules)
    throws ManagementException, RemoteException
```

#### Exceptions

*ManagementException*  
*RemoteException*

---

### **getAllWorkSubstituteRules() Method**

```
public WorkSubstituteRule[] getAllWorkSubstituteRules()
    throws ManagementException, RemoteException
```

## Exceptions

*ManagementException*

*RemoteException*

---

## getWorkSubstitute(String, Date) Method

```
public String getWorkSubstitute(String user,  
                                Date date)  
    throws ManagementException, RemoteException
```

## Exceptions

*ManagementException*

*RemoteException*

---

## getWorkSubstituteRule(String) Method

```
public WorkSubstituteRule getWorkSubstituteRule(String ruleId)  
    throws ManagementException, RemoteException
```

## Exceptions

*ManagementException*

*RemoteException*

---

## getWorkSubstituteRulesWithSource(String) Method

```
public WorkSubstituteRule[] getWorkSubstituteRulesWithSource(String source)  
    throws ManagementException, RemoteException
```

## Exceptions

*ManagementException*

*RemoteException*

---

## getWorkSubstituteRulesWithTarget(String) Method

```
public WorkSubstituteRule[] getWorkSubstituteRulesWithTarget(String target)  
    throws ManagementException, RemoteException
```

## Exceptions

*ManagementException*

*RemoteException*

---



## **removeWorkSubstituteRules(String[]) Method**

```
public void removeWorkSubstituteRules(String[] workSubstituteRuleIds)
    throws ManagementException, RemoteException
```

### **Exceptions**

*ManagementException*

*RemoteException*

---

## **updateWorkSubstituteRules(WorkSubstituteRule[]) Method**

```
public void updateWorkSubstituteRules(WorkSubstituteRule[] rules)
    throws ManagementException, RemoteException
```

### **Exceptions**

*ManagementException*

*RemoteException*

# WorkSubstituteRule Class

*public final class WorkSubstituteRule*

extends Object  
implements Serializable

This class provide information about a workSubstituteRule. A rule is composed of: an id that uniquely identifies the rule an effectiveDate that specifies when the rule takes effect. Null if is always effective. an expiryDate that specifies when the rule expires. Null if never expires. a source that is the user/group to be substituted a target that is the substitute user/group. the source and the target must be both a user or both a group. when a task is assigned to a user, the worklist look for an eventual substitute. The rules can be chained.

---

## Hierarchy

Object  
WorkSubstituteRule

## All Implemented Interfaces

Serializable

---

## Field

### Summary

public  
volatile int **hashCode**  
int

## Constructor

### Summary

*WorkSubstituteRule()*

*WorkSubstituteRule(String workSubstituteRuleId, Date effectiveDate, Date expiryDate, String source, String target)*

## Method

### Summary

public  
boolean **equals**(Object other)

```

public
    Date getEffectiveDate()
public
    Date getExpiryDate()
public
    String getSource()
public
    String getTarget()
public
    String getWorkSubstituteRuleId()
public
    int hashCode()
public
    void setEffectiveDate(Date effectiveDate)
public
    void setExpiryDate(Date expiryDate)
public
    void setSource(String source)
public
    void setTarget(String target)

```

#### Methods from class `java.lang.Object`

`clone`, `finalize`, `getClass`, `notify`, `notifyAll`, `toString`, `wait`, `wait`, `wait`

## Field Detail

### hashCode

```
public volatile int hashCode
```

## Constructor Detail

### WorkSubstituteRule

```
public WorkSubstituteRule()
```

---

### WorkSubstituteRule

```

public WorkSubstituteRule(String workSubstituteRuleId,
                          Date effectiveDate,
                          Date expiryDate,
                          String source,
                          String target)

```

## Method Detail

### **equals(Object) Method**

```
public boolean equals(Object other)
```

#### *Overrides*

```
Object.equals(Object)
```

---

### **getEffectiveDate() Method**

```
public Date getEffectiveDate()
```

---

### **getExpiryDate() Method**

```
public Date getExpiryDate()
```

---

### **getSource() Method**

```
public String getSource()
```

---

### **getTarget() Method**

```
public String getTarget()
```

---

### **getWorkSubstituteRuleId() Method**

```
public String getWorkSubstituteRuleId()
```

---

### **hashCode() Method**

```
public int hashCode()
```

#### *Overrides*

```
Object.hashCode()
```

---

### **setEffectiveDate(Date) Method**

```
public void setEffectiveDate(Date effectiveDate)
```

---

## **setExpiryDate(Date) Method**

```
public void setExpiryDate(Date expiryDate)
```

---

## **setSource(String) Method**

```
public void setSource(String source)
```

---

## **setTarget(String) Method**

```
public void setTarget(String target)
```

# **com.bea.wlxt Package**

## **Interface Summary**

XT

XTPool

# XT Interface

*public interface XT*

## Method Summary

```

public
    void addMFLDocument(URL mflURI)
        Add a MFL document to the cache.

public
    void addMFLDocument(URL mflURI, InputSource mflInput)
        Add a MFL document to the cache.

public
    void addMFLDocument(URL mflURI, InputStream mflInput)
        Add a MFL document to the cache.

public
    void addMFLDocument(URL mflURI, Reader mflInput)

public
    void addStylesheet(URL stylesheetURI)
        Add a single stylesheet to the cache.

public
    void addStylesheet(URL stylesheetURI, InputSource stylesheetInput)
        Add a single stylesheet to the cache.

public
    void clearMFLCache()

public
    void init(Properties properties)
        Initialize the WLXT with a Properties object.

public
    void init(String filename)
        Initialize WLXT with a Properties file.

public
Document parse(URL mflURI, InputStream binaryInput, PrintWriter debug)
    Parse binary data given in binaryInput argument using the MFL document specified by
    mflURI.

public
Document parse(URL mflURI, InputStream binaryInput, String dtdSystemId, PrintWriter
    debug)
    Parse binary data given in binaryInput argument using the MFL document specified by
    mflURI.

public
Document parse(URL mflURI, InputStream binaryInput, String defaultNamespace, String
    schemaLocation, PrintWriter debug)

```

## Java Class Reference

Parse binary data given in `binaryInput` argument using the MFL document specified by `mflURI`.

```
public
    void reinit(Properties properties)
        Reinitialize the entire cache.

public
    void reinitStylesheet(URL stylesheetURI)
        Reinitialize a single stylesheet.

public
    void serialize(URL mflURI, Document document, OutputStream binaryOutput,
        PrintWriter debug)
        Serialize the XML data (given in document argument) to the binary format specified by
        the MFL document (mflURI).

public
    void serialize(URL mflURI, InputStream xmlInput, OutputStream binaryOutput)
        Serialize the XML data (given in xmlInput argument) to the binary format specified by
        the MFL document (mflURI).

public
    void serialize(URL mflURI, InputSource xmlInput, OutputStream binaryOutput)
        Serialize the XML data (given in xmlInput argument) to the binary format specified by
        the MFL document (mflURI).

public
    void serialize(URL mflURI, InputStream xmlInput, OutputStream binaryOutput,
        PrintWriter debug)
        Serialize the XML data (given in xmlInput argument) to the binary format specified by
        the MFL document (mflURI).

public
    void serialize(URL mflURI, InputSource xmlInput, OutputStream binaryOutput,
        PrintWriter debug)
        Serialize the XML data (given in xmlInput argument) to the binary format specified by
        the MFL document (mflURI).

public
    void transform(InputSource xmlInput, OutputStream out, URL stylesheetURI)
        Perform an XSLT transformation on the XML input argument using the specified
        stylesheet.

public
    void transform(Document document, OutputStream out, URL stylesheetURI)
        Perform an XSLT transformation on the document argument using the specified
        stylesheet.

public
Document transform(InputSource xmlInput, URL stylesheetURI)
        Perform an XSLT transformation on the document argument using the specified
        stylesheet.

public
Document transform(Document document, URL stylesheetURI)
        Perform an XSLT transformation on the document argument using the specified
        stylesheet.
```



## Method Detail

### addMFLDocument(URL) Method

#### *DEPRECATED*

```
public void addMFLDocument(URL mflURI)
    throws Exception
```

Add a MFL document to the cache. This method causes the specified MFL document to be parsed and added to the WLXT object's cache. If a MFL Document has previously been added using this URI, the previous document is replaced.

#### Parameters

*mflURI*

The unique resource identifier for the MFL document

#### Exceptions

*Exception*

#### Related Topics

URL

---

### addMFLDocument(URL, InputSource) Method

#### *DEPRECATED*

```
public void addMFLDocument(URL mflURI,
    InputSource mflInput)
    throws Exception
```

Add a MFL document to the cache. The user can supply the InputSource for the content of the MFL document. This method causes the specified MFL document to be parsed and added to the WLXT object's cache. If a MFL Document has previously been added using this URI, the previous document is replaced.

#### Parameters

*mflURI*

The unique resource identifier for the MFL document

*mflInput*

The InputSource for the content of the MFL document

## Exceptions

*Exception*

## Related Topics

URL

---

## addMFLDocument(URL, InputStream) Method

### ***DEPRECATED***

```
public void addMFLDocument(URL mflURI,  
                           InputStream mflInput)  
    throws Exception
```

Add a MFL document to the cache. The user can supply the InputSource for the content of the MFL document. This method causes the specified MFL document to be parsed and added to the WLXT object's cache. If a MFL Document has previously been added using this URI, the previous document is replaced.

## Parameters

*mflURI*

The unique resource identifier for the MFL document

*mflInput*

The InputSource for the content of the MFL document

## Exceptions

*Exception*

## Related Topics

URL

---

## addMFLDocument(URL, Reader) Method

```
public void addMFLDocument(URL mflURI,  
                           Reader mflInput)  
    throws Exception
```

## Exceptions

*Exception*

---

## addStylesheet(URL) Method

### *DEPRECATED*

```
public void addStylesheet(URL stylesheetURI)
    throws Exception
```

Add a single stylesheet to the cache. This method causes the specified stylesheet to be added into the XSLTWrapper object's cache.

### Parameters

*stylesheetURI*

The unique resource identifier for the stylesheet

### Exceptions

*Exception*

### Related Topics

URL

---

## addStylesheet(URL, InputSource) Method

### *DEPRECATED*

```
public void addStylesheet(URL stylesheetURI,
    InputSource stylesheetInput)
    throws Exception
```

Add a single stylesheet to the cache. The user can supply the InputSource for the content of the stylesheet. This method causes the specified stylesheet to be added into the XSLTWrapper object's cache.

### Parameters

*stylesheetURI*

The unique resource identifier for the stylesheet

*stylesheetInput*

The InputSource for the stylesheet

### Exceptions

*Exception*

### Related Topics

URL

---

## clearMFLCache() Method

```
public void clearMFLCache()
```

---

## init(Properties) Method

**DEPRECATED**

```
public void init(Properties properties)
    throws Exception
```

Initialize the WLXT with a Properties object.

### Parameters

*properties*  
Properties object containing the property 'WLXT.stylesheets' and/or 'WLXT.MFLDocuments'.

### Exceptions

*Exception*

### Related Topics

Properties

---

## init(String) Method

**DEPRECATED**

```
public void init(String filename)
    throws Exception
```

Initialize WLXT with a Properties file. The properties file should contain the property 'WLXT.stylesheets' and/or 'WLXT.MFLDocuments'. The stylesheets listed in the 'WLXT.stylesheets' property are precompiled and cached internally. The MFL documents listed in the 'WLXT.MFLDocument' property are also precompiled and cached internally.

### Parameters

*filename*  
The name of the properties file

### Exceptions

*Exception*

---

## parse(URL, InputStream, PrintWriter) Method

### *DEPRECATED*

```
public Document parse(URL mflURI,
                      InputStream binaryInput,
                      PrintWriter debug)
    throws Exception
```

Parse binary data given in binaryInput argument using the MFL document specified by mflURI. The resulting DOM object is returned to the caller.

No DTD or schema reference is output in the resulting XML.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*mflURI*

URL to MFL document that should be used to parse binary data

*binaryInput*

InputStream of binary data that should be converted to XML

*debug*

PrintWriter to be used to display parse debug information. May be null to disable debug information.

### Exceptions

*Exception*

---

## parse(URL, InputStream, String, PrintWriter) Method

### *DEPRECATED*

```
public Document parse(URL mflURI,
                      InputStream binaryInput,
                      String dtdSystemId,
                      PrintWriter debug)
    throws Exception
```

Parse binary data given in binaryInput argument using the MFL document specified by mflURI. The resulting DOM object is returned to the caller.

This method has a system id parameter that allows the user to specify the document type definition for the doctype element of the resulting XML.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

**Parameters***mflURI*

URL to MFL document that should be used to parse binary data

*binaryInput*

InputStream of binary data that should be converted to XML

*dtdSystemId*

The system id of the Document Type Definition. This id will be output in the DOCTYPE element of the resulting XML.

*debug*

PrintWriter to be used to display parse debug information. May be null to disable debug information.

**Exceptions***Exception*

---

**parse(URL, InputStream, String, String, PrintWriter) Method****DEPRECATED**

```

public Document parse(URL mflURI,
                      InputStream binaryInput,
                      String defaultNamespace,
                      String schemaLocation,
                      PrintWriter debug)
    throws Exception

```

Parse binary data given in binaryInput argument using the MFL document specified by mflURI. The resulting DOM object is returned to the caller.

This method has default namespace parameter that causes the resulting XML to output a default namespace in the root node of the document. This method also has a schema location parameter cause the resulting XML to reference a schema as its content model.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

**Parameters***mflURI*

URL to MFL document that should be used to parse binary data

*binaryInput*

InputStream of binary data that should be converted to XML

*defaultNamespace*

can be null, if not null causes a default namespace to be declared in the root node of the resulting XML

*schemaLocation*

causes a reference to the specified schema to be output in the root node of the resulting XML

*debug*

PrintWriter to be used to display parse debug information. May be null to disable debug information.

## Exceptions

*Exception*

---

## reinit(Properties) Method

**DEPRECATED**

```
public void reinit(Properties properties)
    throws Exception
```

Reinitialize the entire cache. This method causes all stylesheets and MFL documents to be reloaded into the cache. The properties object should contain the property 'WLXT.stylesheets' and/or 'WLXT.MFLDocuments'.

## Parameters

*properties*

Properties object containing the property WLXT.stylesheets and/or WLXT.MFLDocuments

## Exceptions

*Exception*

---

## reinitStylesheet(URL) Method

**DEPRECATED**

```
public void reinitStylesheet(URL stylesheetURI)
    throws Exception
```

Reinitialize a single stylesheet. This method causes the specified stylesheet to be reloaded into the cache.

## Parameters

*stylesheetURI*

The unique resource identifier for the stylesheet

## Exceptions

*Exception*

## Related Topics

## serialize(URL, Document, OutputStream, PrintWriter) Method

### DEPRECATED

```
public void serialize(URL mflURI,
                     Document document,
                     OutputStream binaryOutput,
                     PrintWriter debug)
    throws Exception
```

Serialize the XML data (given in document argument) to the binary format specified by the MFL document (mflURI). Each node of XML data is converted to the binary format specified by the MFL FieldFormat of the identical name. The order and grouping of the XML nodes must match the order and sequence specified in the MFL document.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*mflURI*

URL to MFL document that should be used to convert the XML data to the binary format

*document*

XML data that is to be converted to a binary format

*binaryOutput*

OutputStream that is to contain the binary data

*debug*

PrintWriter to be used to display (or log) debug information. May be null to disable debug information.

### Exceptions

*Exception*

---

## serialize(URL, InputStream, OutputStream) Method

### DEPRECATED

```
public void serialize(URL mflURI,
                     InputStream xmlInput,
                     OutputStream binaryOutput)
    throws Exception
```

Serialize the XML data (given in xmlInput argument) to the binary format specified by the MFL document (mflURI). Each node of XML data is converted to the binary format specified by the MFL FieldFormat of the identical name. The order and grouping of the XML nodes must match the order and sequence specified in the



MFL document.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*mflURI*

URL to MFL document that should be used to convert the XML data to the binary format

*xmlInput*

XML data that is to be converted to a binary format

*binaryOutput*

OutputStream that is to contain the binary data

### Exceptions

*Exception*

---

## serialize(URL, InputSource, OutputStream) Method

### DEPRECATED

```
public void serialize(URL mflURI,
                     InputSource xmlInput,
                     OutputStream binaryOutput)
    throws Exception
```

Serialize the XML data (given in xmlInput argument) to the binary format specified by the MFL document (mflURI). Each node of XML data is converted to the binary format specified by the MFL FieldFormat of the identical name. The order and grouping of the XML nodes must match the order and sequence specified in the MFL document.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*mflURI*

URL to MFL document that should be used to convert the XML data to the binary format

*xmlInput*

XML data that is to be converted to a binary format

*binaryOutput*

OutputStream that is to contain the binary data

### Exceptions

*Exception*

---

**serialize(URL, InputStream, OutputStream, PrintWriter) Method*****DEPRECATED***

```
public void serialize(URL mflURI,
                     InputStream xmlInput,
                     OutputStream binaryOutput,
                     PrintWriter debug)
    throws Exception
```

Serialize the XML data (given in xmlInput argument) to the binary format specified by the MFL document (mflURI). Each node of XML data is converted to the binary format specified by the MFL FieldFormat of the identical name. The order and grouping of the XML nodes must match the order and sequence specified in the MFL document.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

**Parameters**

*mflURI*

URL to MFL document that should be used to convert the XML data to the binary format

*xmlInput*

XML data that is to be converted to a binary format

*binaryOutput*

OutputStream that is to contain the binary data

*debug*

PrintWriter to be used to display (or log) debug information. May be null to disable debug information.

**Exceptions**

*Exception*

---

**serialize(URL, InputSource, OutputStream, PrintWriter) Method*****DEPRECATED***

```
public void serialize(URL mflURI,
                     InputSource xmlInput,
                     OutputStream binaryOutput,
                     PrintWriter debug)
    throws Exception
```

Serialize the XML data (given in xmlInput argument) to the binary format specified by the MFL document (mflURI). Each node of XML data is converted to the binary format specified by the MFL FieldFormat of the identical name. The order and grouping of the XML nodes must match the order and sequence specified in the MFL document.

If the MFL document is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*mflURI*

URL to MFL document that should be used to convert the XML data to the binary format

*xmlInput*

XML data that is to be converted to a binary format

*binaryOutput*

OutputStream that is to contain the binary data

*debug*

PrintWriter to be used to display (or log) debug information. May be null to disable debug information.

### Exceptions

*Exception*

---

## transform(InputSource, OutputStream, URL) Method

### DEPRECATED

```
public void transform(InputSource xmlInput,
                     OutputStream out,
                     URL stylesheetURI)
    throws Exception
```

Perform an XSLT transformation on the XML input argument using the specified stylesheet. The result of the transformation is written to the output stream parameter. If the stylesheet is not found in the cache, then it will be automatically processed and added to the cache.

### Parameters

*xmlInput*

XML input to XSLT transformation

*out*

Output stream where the result of transformation is written.

*stylesheetURI*

The unique resource identifier for the stylesheet to use for the transformation

### Exceptions

*Exception*

### Related Topics

InputSource  
URL

---

## transform(Document, OutputStream, URL) Method

### *DEPRECATED*

```
public void transform(Document document,  
                      OutputStream out,  
                      URL stylesheetURI)  
    throws Exception
```

Perform an XSLT transformation on the document argument using the specified stylesheet. The result of the transformation is put in the result parameter.

### Parameters

*document*

DOM Document object to be transformed

*out*

Output stream where the result of transformation is written.

*stylesheetURI*

The unique resource identifier for the stylesheet to use for the transformation

### Exceptions

*Exception*

### Related Topics

Document  
URL

---

## transform(InputSource, URL) Method

### *DEPRECATED*

```
public Document transform(InputSource xmlInput,  
                          URL stylesheetURI)  
    throws Exception
```

Perform an XSLT transformation on the document argument using the specified stylesheet. The result of the transformation is returned.

**Parameters***xmlInput*

XML input to XSLT transformation

*stylesheetURI*

The unique resource identifier for the stylesheet to use for the transformation

**Returns***document DOM Document object that results from transformation***Exceptions***Exception***Related Topics**

Document

URL

**transform(Document, URL) Method****DEPRECATED**

```
public Document transform(Document document,
                          URL stylesheetURI)
    throws Exception
```

Perform an XSLT transformation on the document argument using the specified stylesheet. The result of the transformation is returned.

**Parameters***document*

DOM Document object to be transformed

*stylesheetURI*

The unique resource identifier for the stylesheet to use for the transformation

**Returns***document DOM Document object that results from transformation***Exceptions***Exception***Related Topics**

Document  
URL

# XTPool Interface

*public interface XTPool*

## Method Summary

```
public
    void flushXT()
        Flush the WLXT pool requiring new objects to be created and all MFL/Stylesheet caches to
        be repopulated.

public
    XT getXT()
        Get a WLXT instance from the pool.

public
    void putXT(XT wlxt)
        Return a WLXT instance to the pool.

public
    void setMaxPoolXT(int maxPool)
        Set the maximum instances of WLXT that will be maintained in the pool.
```

## Method Detail

### flushXT() Method

**DEPRECATED**

```
public void flushXT()
```

Flush the WLXT pool requiring new objects to be created and all MFL/Stylesheet caches to be repopulated.

---

### getXT() Method

**DEPRECATED**

```
public XT getXT()
```

Get a WLXT instance from the pool.

### Returns

*An instance of WLXT.*

---

## putXT(XT) Method

*DEPRECATED*

```
public void putXT(XT wLxt)
```

Return a WLXT instance to the pool.

### Parameters

*wLxt*

The object instance to be placed in the pool.

---

## setMaxPoolXT(int) Method

*DEPRECATED*

```
public void setMaxPoolXT(int maxPool)
```

Set the maximum instances of WLXT that will be maintained in the pool. This value is JVM global.

### Parameters

*maxPool*

The maximum number if WLXT objects that will be retained by the pool.



# com.bea.xml Package

## Interface Summary

`XmlObjectList` Interface to represent lists of `XmlObject` instances returned from a query.

## Class Summary

`XmlObjectList.Factory` Static factory class for creating new instances of `XmlObjectList`

`XmlUtils` Class providing some Utils for Xml Objects.

## Exception Summary

`XmlUtilsException` Exception for Transformation runtime exceptions.

# XmlObjectList Interface

*public interface XmlObjectList*

Interface to represent lists of XmlObject instances returned from a query. Used also as a convenient typed way of collecting XmlObject instances.

## Nested Class Summary

```

    public static final class XmlObjectList.Factory
        Static factory class for creating new instances of XmlObjectList

```

## Method Summary

```

    public void add(XmlObject xml)
        Adds a XmlObject to the end of the list.

    public void addAll(XmlObjectList xmlList)
        Appends the contents of another list to this list.

    public void addAll(XmlObject[] xmlArray)
        Appends an array of XmlObject instances to this list.

    public void clear()
        Clears out the contents of this list.

    public XmlObject get(int index)
        Gets the XmlObject at a specified position in the list.

    public boolean isEmpty()
        Tests if this XmlList is empty.

    public Iterator iterator()
        Returns an Iterator over all the XmlObject instances contained in this list.

    public int size()
        Returns the number of XmlObject instances contained in this list.

    public XmlObject[] toArray()
        Returns a copy of the contents of the XmlList in a newly created array of
        XmlObjects.

```

## Method Detail

### add(XmlObject) Method

```
public void add(XmlObject xml)
```

Adds a XmlObject to the end of the list.

#### Parameters

*xml*

the XmlObject instance to add to the list

---

### addAll(XmlObjectList) Method

```
public void addAll(XmlObjectList xmlList)
```

Appends the contents of another list to this list.

#### Parameters

*xmlList*

the list which is to be appended to this list.

---

### addAll(XmlObject[]) Method

```
public void addAll(XmlObject[] xmlArray)
```

Appends an array of XmlObject instances to this list.

#### Parameters

*xmlArray*

the array which is to be appended to this list.

---

### clear() Method

```
public void clear()
```

Clears out the contents of this list. After this operation, isEmpty() will return true.

---

### get(int) Method

```
public XmlObject get(int index)
```

Gets the XmlObject at a specified position in the list.

### Parameters

*index*

the specified position

### Returns

*the XmlObject at the position in this list.*

---

## isEmpty() Method

```
public boolean isEmpty()
```

Tests if this XmlList is empty. This is equivalent to `size() == 0`.

### Returns

*boolean indicating if this list is empty.*

---

## iterator() Method

```
public Iterator iterator()
```

Returns an Iterator over all the XmlObject instances contained in this list. This iterator will throw an UnsupportedOperationException in response to its remove method.

### Returns

*Iterator over the contents of this list*

---

## size() Method

```
public int size()
```

Returns the number of XmlObject instances contained in this list.

### Returns

*int size of the list.*

---

## toArray() Method

```
public XmlObject[] toArray()
```

## Java Class Reference

Returns a copy of the contents of the XmlList in a newly created array of XmlObjects.

### **Returns**

*array of XmlObject instances which is a copy of the contents of the list.*

# XmlObjectList.Factory Class

*public static final class XmlObjectList.Factory*

extends Object

Static factory class for creating new instances of XmlObjectList

---

## Hierarchy

Object  
  XmlObjectList.Factory

## Enclosing interface

XmlObjectList

---

## Constructor Summary

*XmlObjectList.Factory()*

## Method Summary

```
    public
    static newInstance()
XmlObjectList      creates a new empty XmlObjectList instance.
    public
    static newInstance(XmlObjectList list)
XmlObjectList      creates a new XmlObjectList instance, filled with the elements of the given
                    list
    public
    static newInstance(XmlObject[] xmlObjects)
XmlObjectList      creates a new XmlObjectList instance given an array of XmlObjects.
```

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### XmlObjectList.Factory

```
public XmlObjectList.Factory()
```

## Method Detail

### newInstance() Method

```
public static XmlObjectList newInstance()
```

creates a new empty XmlObjectList instance.

#### Returns

*a new empty XmlObjectList instance*

---

### newInstance(XmlObjectList) Method

```
public static XmlObjectList newInstance(XmlObjectList list)
```

creates a new XmlObjectList instance, filled with the elements of the given list

#### Returns

*a new XmlObjectList instance.*

---

### newInstance(XmlObject[]) Method

```
public static XmlObjectList newInstance(XmlObject[] xmlObjects)
```

creates a new XmlObjectList instance given an array of XmlObjects.

#### Returns

*a new XmlObjectList instance.*

# XmlUtils Class

*public abstract class XmlUtils*

extends Object

Class providing some Utils for Xml Objects.

---

## Hierarchy

Object  
XmlUtils

---

## Constructor Summary

*XmlUtils()*

default  
constructor

## Method Summary

public  
static ***newInstance()***  
XmlUtils returns a new implementation instance  
public  
abstract ***validate(XmlObject xmlObject, String systemId, EntityResolver resolver)***  
boolean Schema validates an XmlObject using the given EntityResolver to find the schema.

## Methods from class java.lang.Object

clone, equals, finalize, getClass, hashCode, notify, notifyAll,  
toString, wait, wait, wait

## Constructor Detail

### XmlUtils

**protected XmlUtils()**

default constructor



## Method Detail

### **newInstance() Method**

```
public static XmlUtils newInstance()  
    throws XmlUtilsException
```

returns a new implementation instance

#### **Returns**

*An instance of XmlUtils.*

#### **Exceptions**

*XmlUtilsException*

---

### **validate(XmlObject, String, EntityResolver) Method**

```
public abstract boolean validate(XmlObject xmlObject,  
                                String systemId,  
                                EntityResolver resolver)  
    throws XmlUtilsException
```

Schema validates an `XmlObject` using the given `EntityResolver` to find the schema. The schema is resolved by system id. The system id can be specified either as parameter or using the `xsi:schemaLocation` or `xsi:noNamespaceSchemaLocation` attribute.

#### **Parameters**

*xmlObject*

The XML instance to validate

*systemId*

Expected system id. The system id must be present if the xml document does not specify an `xsi:schemaLocation` or `xsi:noNamespaceSchemaLocation`

*resolver*

The entity resolver used to get the schema.

#### **Exceptions**

*XmlUtilsException*

# XmlUtilsException Class

*public class XmlUtilsException*

extends Exception

Exception for Transformation runtime exceptions.

---

## Hierarchy

```

Object
  Throwable
    Exception
      XmlUtilsException
  
```

## All Implemented Interfaces

```

Serializable
  
```

---

## Field Summary

```

protected int
    _code
        the code

protected String
    _message
        a possible user message

    protected
    static _messages
final Properties        exception messages

protected Object
    _params
        the parameters

    protected
    static DEFAULT_MESSAGE
final String        String

    public
    static INSTANCE_CREATION
final int        int

    public
    static INTERNAL_ERROR
final int        int

    public
    static IO_ERROR
final int
  
```

```

        int

    public
    static MISSING_SYSTEM_ID
    final int
        int

    public
    static NO_CODE_MSG
    final int
        error codes

    public
    static XML_ERROR
    final int
        int

```

## Constructor Summary

***XmlUtilsException**(int code, Object param1, Object param2, Throwable cause)*

***XmlUtilsException**(int code, Object param1, Object param2)*

***XmlUtilsException**(int code, Object param, Throwable cause)*

***XmlUtilsException**(int code, Object param)*

***XmlUtilsException**(int code, Object params, Throwable cause)*

***XmlUtilsException**(int code, Object params)*

***XmlUtilsException**(int code, Throwable cause)*

***XmlUtilsException**(int code)*

constructors for system defined exceptions

***XmlUtilsException**(String message, Object param1, Object param2, Throwable cause)*

***XmlUtilsException**(String message, Object param1, Object param2)*

*XmlUtilsException(String message, Object param, Throwable cause)*

*XmlUtilsException(String message, Object param)*

*XmlUtilsException(String message, Object params, Throwable cause)*

*XmlUtilsException(String message, Object params)*

*XmlUtilsException(String message, Throwable cause)*

*XmlUtilsException(String message)*

constructors for user defined exceptions

## Method Summary

```
public
String getMessage()
    exception
    message
```

### Methods from `java.lang.Throwable`

```
fillInStackTrace, getCause, getLocalizedMessage, getStackTrace,
initCause, printStackTrace, printStackTrace, printStackTrace,
setStackTrace, toString
```

### Methods from class `java.lang.Object`

```
clone, equals, finalize, getClass, hashCode, notify, notifyAll,
toString, wait, wait, wait
```

## Field Detail

### **`_code`**

```
protected int _code
```

the code

**\_message**

```
protected String _message
```

a possible user message

---

**\_messages**

```
protected static final Properties _messages
```

exception messages

---

**\_params**

```
protected Object _params
```

the parameters

---

**DEFAULT\_MESSAGE**

```
protected static final String DEFAULT_MESSAGE
```

---

**INSTANCE\_CREATION**

```
public static final int INSTANCE_CREATION
```

---

**INTERNAL\_ERROR**

```
public static final int INTERNAL_ERROR
```

---

**IO\_ERROR**

```
public static final int IO_ERROR
```

---

**MISSING\_SYSTEM\_ID**

```
public static final int MISSING_SYSTEM_ID
```

---

**NO\_CODE\_MSG**

```
public static final int NO_CODE_MSG
```

error codes

---

## XML\_ERROR

```
public static final int XML_ERROR
```

## Constructor Detail

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object param1,  
                        Object param2,  
                        Throwable cause)
```

---

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object param1,  
                        Object param2)
```

---

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object param,  
                        Throwable cause)
```

---

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object param)
```

---

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object[] params,  
                        Throwable cause)
```

---

### XmlUtilsException

```
public XmlUtilsException(int code,  
                        Object[] params)
```

---

## XmlUtilsException

```
public XmlUtilsException(int code,  
                          Throwable cause)
```

---

## XmlUtilsException

```
public XmlUtilsException(int code)
```

constructors for system defined exceptions

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object param1,  
                          Object param2,  
                          Throwable cause)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object param1,  
                          Object param2)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object param,  
                          Throwable cause)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object param)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object[] params,  
                          Throwable cause)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Object[] params)
```

## XmlUtilsException

```
public XmlUtilsException(String message,  
                          Throwable cause)
```

---

## XmlUtilsException

```
public XmlUtilsException(String message)
```

constructors for user defined exceptions

## Method Detail

### getMessage() Method

```
public String getMessage()
```

exception message

#### *Overrides*

Throwable.getMessage()