| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SOAPMessage.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/soap/SOAPHeaderElement.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/soap/SOAPPart.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/soap/SOAPMessage.html)    [**NO FRAMES**](http://docs.google.com/SOAPMessage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

## **javax.xml.soap**

Class SOAPMessage

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.xml.soap.SOAPMessage**

public abstract class **SOAPMessage**extends [Object](http://docs.google.com/java/lang/Object.html)

The root class for all SOAP messages. As transmitted on the "wire", a SOAP message is an XML document or a MIME message whose first body part is an XML/SOAP document.

A SOAPMessage object consists of a SOAP part and optionally one or more attachment parts. The SOAP part for a SOAPMessage object is a SOAPPart object, which contains information used for message routing and identification, and which can contain application-specific content. All data in the SOAP Part of a message must be in XML format.

A new SOAPMessage object contains the following by default:

* A SOAPPart object
* A SOAPEnvelope object
* A SOAPBody object
* A SOAPHeader object

The SOAP part of a message can be retrieved by calling the method SOAPMessage.getSOAPPart(). The SOAPEnvelope object is retrieved from the SOAPPart object, and the SOAPEnvelope object is used to retrieve the SOAPBody and SOAPHeader objects.

SOAPPart sp = message.getSOAPPart();  
 SOAPEnvelope se = sp.getEnvelope();  
 SOAPBody sb = se.getBody();  
 SOAPHeader sh = se.getHeader();

In addition to the mandatory SOAPPart object, a SOAPMessage object may contain zero or more AttachmentPart objects, each of which contains application-specific data. The SOAPMessage interface provides methods for creating AttachmentPart objects and also for adding them to a SOAPMessage object. A party that has received a SOAPMessage object can examine its contents by retrieving individual attachment parts.

Unlike the rest of a SOAP message, an attachment is not required to be in XML format and can therefore be anything from simple text to an image file. Consequently, any message content that is not in XML format must be in an AttachmentPart object.

A MessageFactory object may create SOAPMessage objects with behavior that is specialized to a particular implementation or application of SAAJ. For instance, a MessageFactory object may produce SOAPMessage objects that conform to a particular Profile such as ebXML. In this case a MessageFactory object might produce SOAPMessage objects that are initialized with ebXML headers.

In order to ensure backward source compatibility, methods that are added to this class after version 1.1 of the SAAJ specification are all concrete instead of abstract and they all have default implementations. Unless otherwise noted in the JavaDocs for those methods the default implementations simply throw an UnsupportedOperationException and the SAAJ implementation code must override them with methods that provide the specified behavior. Legacy client code does not have this restriction, however, so long as there is no claim made that it conforms to some later version of the specification than it was originally written for. A legacy class that extends the SOAPMessage class can be compiled and/or run against succeeding versions of the SAAJ API without modification. If such a class was correctly implemented then it will continue to behave correctly relative to the version of the specification against which it was written.

**See Also:**[MessageFactory](http://docs.google.com/javax/xml/soap/MessageFactory.html), [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html)

| **Field Summary** | |
| --- | --- |
| static [String](http://docs.google.com/java/lang/String.html) | [**CHARACTER\_SET\_ENCODING**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#CHARACTER_SET_ENCODING)            Specifies the character type encoding for the SOAP Message. |
| static [String](http://docs.google.com/java/lang/String.html) | [**WRITE\_XML\_DECLARATION**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#WRITE_XML_DECLARATION)            Specifies whether the SOAP Message will contain an XML declaration when it is sent. |

| **Constructor Summary** | |
| --- | --- |
| [**SOAPMessage**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#SOAPMessage())() |

| **Method Summary** | |
| --- | --- |
| abstract  void | [**addAttachmentPart**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#addAttachmentPart(javax.xml.soap.AttachmentPart))([AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) AttachmentPart)            Adds the given AttachmentPart object to this SOAPMessage object. |
| abstract  int | [**countAttachments**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#countAttachments())()            Gets a count of the number of attachments in this message. |
| abstract  [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) | [**createAttachmentPart**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#createAttachmentPart())()            Creates a new empty AttachmentPart object. |
| [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) | [**createAttachmentPart**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#createAttachmentPart(javax.activation.DataHandler))([DataHandler](http://docs.google.com/javax/activation/DataHandler.html) dataHandler)            Creates an AttachmentPart object and populates it using the given DataHandler object. |
| [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) | [**createAttachmentPart**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#createAttachmentPart(java.lang.Object,%20java.lang.String))([Object](http://docs.google.com/java/lang/Object.html) content, [String](http://docs.google.com/java/lang/String.html) contentType)            Creates an AttachmentPart object and populates it with the specified data of the specified content type. |
| abstract  [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) | [**getAttachment**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getAttachment(javax.xml.soap.SOAPElement))([SOAPElement](http://docs.google.com/javax/xml/soap/SOAPElement.html) element)            Returns an AttachmentPart object that is associated with an attachment that is referenced by this SOAPElement or null if no such attachment exists. |
| abstract  [Iterator](http://docs.google.com/java/util/Iterator.html) | [**getAttachments**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getAttachments())()            Retrieves all the AttachmentPart objects that are part of this SOAPMessage object. |
| abstract  [Iterator](http://docs.google.com/java/util/Iterator.html) | [**getAttachments**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getAttachments(javax.xml.soap.MimeHeaders))([MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) headers)            Retrieves all the AttachmentPart objects that have header entries that match the specified headers. |
| abstract  [String](http://docs.google.com/java/lang/String.html) | [**getContentDescription**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getContentDescription())()            Retrieves a description of this SOAPMessage object's content. |
| abstract  [MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) | [**getMimeHeaders**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getMimeHeaders())()            Returns all the transport-specific MIME headers for this SOAPMessage object in a transport-independent fashion. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getProperty**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getProperty(java.lang.String))([String](http://docs.google.com/java/lang/String.html) property)            Retrieves value of the specified property. |
| [SOAPBody](http://docs.google.com/javax/xml/soap/SOAPBody.html) | [**getSOAPBody**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getSOAPBody())()            Gets the SOAP Body contained in this SOAPMessage object. |
| [SOAPHeader](http://docs.google.com/javax/xml/soap/SOAPHeader.html) | [**getSOAPHeader**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getSOAPHeader())()            Gets the SOAP Header contained in this SOAPMessage object. |
| abstract  [SOAPPart](http://docs.google.com/javax/xml/soap/SOAPPart.html) | [**getSOAPPart**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getSOAPPart())()            Gets the SOAP part of this SOAPMessage object. |
| abstract  void | [**removeAllAttachments**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#removeAllAttachments())()            Removes all AttachmentPart objects that have been added to this SOAPMessage object. |
| abstract  void | [**removeAttachments**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#removeAttachments(javax.xml.soap.MimeHeaders))([MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) headers)            Removes all the AttachmentPart objects that have header entries that match the specified headers. |
| abstract  void | [**saveChanges**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#saveChanges())()            Updates this SOAPMessage object with all the changes that have been made to it. |
| abstract  boolean | [**saveRequired**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#saveRequired())()            Indicates whether this SOAPMessage object needs to have the method saveChanges called on it. |
| abstract  void | [**setContentDescription**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#setContentDescription(java.lang.String))([String](http://docs.google.com/java/lang/String.html) description)            Sets the description of this SOAPMessage object's content with the given description. |
| void | [**setProperty**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#setProperty(java.lang.String,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) property, [Object](http://docs.google.com/java/lang/Object.html) value)            Associates the specified value with the specified property. |
| abstract  void | [**writeTo**](http://docs.google.com/javax/xml/soap/SOAPMessage.html#writeTo(java.io.OutputStream))([OutputStream](http://docs.google.com/java/io/OutputStream.html) out)            Writes this SOAPMessage object to the given output stream. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Field Detail** |
| --- |

### CHARACTER\_SET\_ENCODING

public static final [String](http://docs.google.com/java/lang/String.html) **CHARACTER\_SET\_ENCODING**

Specifies the character type encoding for the SOAP Message. Valid values include "utf-8" and "utf-16". See vendor documentation for additional supported values. The default is "utf-8".

**Since:** SAAJ 1.2 **See Also:**[SOAPMessage.setProperty](http://docs.google.com/javax/xml/soap/SOAPMessage.html#setProperty(java.lang.String,%20java.lang.Object)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.xml.soap.SOAPMessage.CHARACTER_SET_ENCODING)

### WRITE\_XML\_DECLARATION

public static final [String](http://docs.google.com/java/lang/String.html) **WRITE\_XML\_DECLARATION**

Specifies whether the SOAP Message will contain an XML declaration when it is sent. The only valid values are "true" and "false". The default is "false".

**Since:** SAAJ 1.2 **See Also:**[SOAPMessage.setProperty](http://docs.google.com/javax/xml/soap/SOAPMessage.html#setProperty(java.lang.String,%20java.lang.Object)), [Constant Field Values](http://docs.google.com/constant-values.html#javax.xml.soap.SOAPMessage.WRITE_XML_DECLARATION)

| **Constructor Detail** |
| --- |

### SOAPMessage

public **SOAPMessage**()

| **Method Detail** |
| --- |

### setContentDescription

public abstract void **setContentDescription**([String](http://docs.google.com/java/lang/String.html) description)

Sets the description of this SOAPMessage object's content with the given description.

**Parameters:**description - a String describing the content of this message**See Also:**[getContentDescription()](http://docs.google.com/javax/xml/soap/SOAPMessage.html#getContentDescription())

### getContentDescription

public abstract [String](http://docs.google.com/java/lang/String.html) **getContentDescription**()

Retrieves a description of this SOAPMessage object's content.

**Returns:**a String describing the content of this message or null if no description has been set**See Also:**[setContentDescription(java.lang.String)](http://docs.google.com/javax/xml/soap/SOAPMessage.html#setContentDescription(java.lang.String))

### getSOAPPart

public abstract [SOAPPart](http://docs.google.com/javax/xml/soap/SOAPPart.html) **getSOAPPart**()

Gets the SOAP part of this SOAPMessage object.

SOAPMessage object contains one or more attachments, the SOAP Part must be the first MIME body part in the message.

**Returns:**the SOAPPart object for this SOAPMessage object

### getSOAPBody

public [SOAPBody](http://docs.google.com/javax/xml/soap/SOAPBody.html) **getSOAPBody**()  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Gets the SOAP Body contained in this SOAPMessage object.

**Returns:**the SOAPBody object contained by this SOAPMessage object **Throws:** [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if the SOAP Body does not exist or cannot be retrieved**Since:** SAAJ 1.2

### getSOAPHeader

public [SOAPHeader](http://docs.google.com/javax/xml/soap/SOAPHeader.html) **getSOAPHeader**()  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Gets the SOAP Header contained in this SOAPMessage object.

**Returns:**the SOAPHeader object contained by this SOAPMessage object **Throws:** [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if the SOAP Header does not exist or cannot be retrieved**Since:** SAAJ 1.2

### removeAllAttachments

public abstract void **removeAllAttachments**()

Removes all AttachmentPart objects that have been added to this SOAPMessage object.

This method does not touch the SOAP part.

### countAttachments

public abstract int **countAttachments**()

Gets a count of the number of attachments in this message. This count does not include the SOAP part.

**Returns:**the number of AttachmentPart objects that are part of this SOAPMessage object

### getAttachments

public abstract [Iterator](http://docs.google.com/java/util/Iterator.html) **getAttachments**()

Retrieves all the AttachmentPart objects that are part of this SOAPMessage object.

**Returns:**an iterator over all the attachments in this message

### getAttachments

public abstract [Iterator](http://docs.google.com/java/util/Iterator.html) **getAttachments**([MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) headers)

Retrieves all the AttachmentPart objects that have header entries that match the specified headers. Note that a returned attachment could have headers in addition to those specified.

**Parameters:**headers - a MimeHeaders object containing the MIME headers for which to search **Returns:**an iterator over all attachments that have a header that matches one of the given headers

### removeAttachments

public abstract void **removeAttachments**([MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) headers)

Removes all the AttachmentPart objects that have header entries that match the specified headers. Note that the removed attachment could have headers in addition to those specified.

**Parameters:**headers - a MimeHeaders object containing the MIME headers for which to search**Since:** SAAJ 1.3

### getAttachment

public abstract [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) **getAttachment**([SOAPElement](http://docs.google.com/javax/xml/soap/SOAPElement.html) element)  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Returns an AttachmentPart object that is associated with an attachment that is referenced by this SOAPElement or null if no such attachment exists. References can be made via an href attribute as described in [SOAP Messages with Attachments](http://www.w3.org/TR/SOAP-attachments#SOAPReferenceToAttachements), or via a single Text child node containing a URI as described in the WS-I Attachments Profile 1.0 for elements of schema type *ref:swaRef*([ref:swaRef](http://www.ws-i.org/Profiles/AttachmentsProfile-1.0-2004-08-24.html%22)). These two mechanisms must be supported. The support for references via href attribute also implies that this method should also be supported on an element that is an *xop:Include* element ( [XOP](http://www.w3.org/2000/xp/Group/3/06/Attachments/XOP.html)). other reference mechanisms may be supported by individual implementations of this standard. Contact your vendor for details.

**Parameters:**element - The SOAPElement containing the reference to an Attachment **Returns:**the referenced AttachmentPart or null if no such AttachmentPart exists or no reference can be found in this SOAPElement. **Throws:** [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if there is an error in the attempt to access the attachment**Since:** SAAJ 1.3

### addAttachmentPart

public abstract void **addAttachmentPart**([AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) AttachmentPart)

Adds the given AttachmentPart object to this SOAPMessage object. An AttachmentPart object must be created before it can be added to a message.

**Parameters:**AttachmentPart - an AttachmentPart object that is to become part of this SOAPMessage object **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html)

### createAttachmentPart

public abstract [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) **createAttachmentPart**()

Creates a new empty AttachmentPart object. Note that the method addAttachmentPart must be called with this new AttachmentPart object as the parameter in order for it to become an attachment to this SOAPMessage object.

**Returns:**a new AttachmentPart object that can be populated and added to this SOAPMessage object

### createAttachmentPart

public [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) **createAttachmentPart**([DataHandler](http://docs.google.com/javax/activation/DataHandler.html) dataHandler)

Creates an AttachmentPart object and populates it using the given DataHandler object.

**Parameters:**dataHandler - the javax.activation.DataHandler object that will generate the content for this SOAPMessage object **Returns:**a new AttachmentPart object that contains data generated by the given DataHandler object **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if there was a problem with the specified DataHandler object**See Also:**[DataHandler](http://docs.google.com/javax/activation/DataHandler.html), [DataContentHandler](http://docs.google.com/javax/activation/DataContentHandler.html)

### getMimeHeaders

public abstract [MimeHeaders](http://docs.google.com/javax/xml/soap/MimeHeaders.html) **getMimeHeaders**()

Returns all the transport-specific MIME headers for this SOAPMessage object in a transport-independent fashion.

**Returns:**a MimeHeaders object containing the MimeHeader objects

### createAttachmentPart

public [AttachmentPart](http://docs.google.com/javax/xml/soap/AttachmentPart.html) **createAttachmentPart**([Object](http://docs.google.com/java/lang/Object.html) content,  
 [String](http://docs.google.com/java/lang/String.html) contentType)

Creates an AttachmentPart object and populates it with the specified data of the specified content type. The type of the Object should correspond to the value given for the Content-Type.

**Parameters:**content - an Object containing the content for the AttachmentPart object to be createdcontentType - a String object giving the type of content; examples are "text/xml", "text/plain", and "image/jpeg" **Returns:**a new AttachmentPart object that contains the given data **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - may be thrown if the contentType does not match the type of the content object, or if there was no DataContentHandler object for the given content object**See Also:**[DataHandler](http://docs.google.com/javax/activation/DataHandler.html), [DataContentHandler](http://docs.google.com/javax/activation/DataContentHandler.html)

### saveChanges

public abstract void **saveChanges**()  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Updates this SOAPMessage object with all the changes that have been made to it. This method is called automatically when [writeTo(OutputStream)](http://docs.google.com/javax/xml/soap/SOAPMessage.html#writeTo(java.io.OutputStream)) is called. However, if changes are made to a message that was received or to one that has already been sent, the method saveChanges needs to be called explicitly in order to save the changes. The method saveChanges also generates any changes that can be read back (for example, a MessageId in profiles that support a message id). All MIME headers in a message that is created for sending purposes are guaranteed to have valid values only after saveChanges has been called.

In addition, this method marks the point at which the data from all constituent AttachmentPart objects are pulled into the message.

**Throws:** SOAPException - if there was a problem saving changes to this message. [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

### saveRequired

public abstract boolean **saveRequired**()

Indicates whether this SOAPMessage object needs to have the method saveChanges called on it.

**Returns:**true if saveChanges needs to be called; false otherwise.

### writeTo

public abstract void **writeTo**([OutputStream](http://docs.google.com/java/io/OutputStream.html) out)  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html),  
 [IOException](http://docs.google.com/java/io/IOException.html)

Writes this SOAPMessage object to the given output stream. The externalization format is as defined by the SOAP 1.1 with Attachments specification.

If there are no attachments, just an XML stream is written out. For those messages that have attachments, writeTo writes a MIME-encoded byte stream.

Note that this method does not write the transport-specific MIME Headers of the Message

**Parameters:**out - the OutputStream object to which this SOAPMessage object will be written **Throws:** [IOException](http://docs.google.com/java/io/IOException.html) - if an I/O error occurs [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if there was a problem in externalizing this SOAP message

### setProperty

public void **setProperty**([String](http://docs.google.com/java/lang/String.html) property,  
 [Object](http://docs.google.com/java/lang/Object.html) value)  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Associates the specified value with the specified property. If there was already a value associated with this property, the old value is replaced.

The valid property names include [WRITE\_XML\_DECLARATION](http://docs.google.com/javax/xml/soap/SOAPMessage.html#WRITE_XML_DECLARATION) and [CHARACTER\_SET\_ENCODING](http://docs.google.com/javax/xml/soap/SOAPMessage.html#CHARACTER_SET_ENCODING). All of these standard SAAJ properties are prefixed by "javax.xml.soap". Vendors may also add implementation specific properties. These properties must be prefixed with package names that are unique to the vendor.

Setting the property WRITE\_XML\_DECLARATION to "true" will cause an XML Declaration to be written out at the start of the SOAP message. The default value of "false" suppresses this declaration.

The property CHARACTER\_SET\_ENCODING defaults to the value "utf-8" which causes the SOAP message to be encoded using UTF-8. Setting CHARACTER\_SET\_ENCODING to "utf-16" causes the SOAP message to be encoded using UTF-16.

Some implementations may allow encodings in addition to UTF-8 and UTF-16. Refer to your vendor's documentation for details.

**Parameters:**property - the property with which the specified value is to be associated.value - the value to be associated with the specified property **Throws:** [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if the property name is not recognized.**Since:** SAAJ 1.2

### getProperty

public [Object](http://docs.google.com/java/lang/Object.html) **getProperty**([String](http://docs.google.com/java/lang/String.html) property)  
 throws [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html)

Retrieves value of the specified property.

**Parameters:**property - the name of the property to retrieve **Returns:**the value associated with the named property or null if no such property exists. **Throws:** [SOAPException](http://docs.google.com/javax/xml/soap/SOAPException.html) - if the property name is not recognized.**Since:** SAAJ 1.2

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SOAPMessage.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/xml/soap/SOAPHeaderElement.html)   [**NEXT CLASS**](http://docs.google.com/javax/xml/soap/SOAPPart.html) | [**FRAMES**](http://docs.google.com/index.html?javax/xml/soap/SOAPMessage.html)    [**NO FRAMES**](http://docs.google.com/SOAPMessage.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#26in1rg) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).