BEA WebLogic Network Gatekeeper™

API Description for Extended Web Services

Version 1.0
Revised: March 14, 2005

Copyright

Restricted Rights Legend

Trademarks or Service Marks

Contents

5. Call User InterAction

6. Content-Based Charging

7. Messaging

9. Subscriber Profile

10.User Location

11.User Status

CHAPTER 1

Introduction and Roadmap

The following sections describe the audience for and organization of this document:

Document Scope and Audience

The purpose of this guide is to describe the interfaces for the Extended Web Services interfaces exposed by the WebLogic Network Gatekeeper. The reader of this document is assumed to have through understanding of web services and Web Services Description Language. Basic understanding of telecommunication technology is also assumed.

For basic knowledge about BEA WebLogic Network Gatekeeper and the functions it provides, refer to the Product Description.

For basic knowledge about how to use the Extended Web services interfaces, refer to the Developer's Guide for the Extended Web Services.

All descriptions are based on the WSDL definitions of the Web Service for the different service capabilities offered by WebLogic Network Gatekeeper.

Guide to this Document

- Chapter 1, "Introduction and Roadmap," informs you about the structure and contents of this document, the used writing conventions, and related documentation.
- Chapter 2, "Extended Web Services Interface," gives general information on the Extended Web Services interfaces.
- Chapter 3, "Access," gives detailed information on the Access service capability.
- Chapter 4, "Call Control," gives detailed information on the Call Control service capability.
- Chapter 5, "Call User InterAction," gives detailed information on the Call User interaction service capability.
- Chapter 6, "Content-Based Charging," gives detailed information on the Content based charging service capability.
- Chapter 7, "Messaging," gives detailed information on the Messaging service capability.
- Chapter 8, "Messaging User Interaction," gives detailed information on the Messaging User Interaction service capability.
- Chapter 9, "Subscriber Profile," gives detailed information on the Subscriber Profile service capability.
- Chapter 10, "User Location," gives detailed information on the User Location service capability.

- Chapter 11, “User Status,” gives detailed information on the User Status service capability.

CHAPTER 2

Extended Web Services Interface

The following section provides general information about the Extended Web Services interfaces.

About the Extended Web Services interface

The Extended Web Service interface offers an very high level abstraction of the of the OSA/Parlay interface, for rapid application development.

The interface of each service capability offered by the Extended Web Service is defined in a set of packages. Each package consists of two WSDL files, one defining the operations an application invokes on Web Service and one defining the operations the WebLogic Network Gatekeeper invokes on the application. The latter is referred to as listener interfaces. The packages are:

Package	Description
Access	Package with methods for handling the access service capability.
Call Control	Package with Web Services containing methods for handling the call control service capability.
Call User Interaction	Package with Web Services containing methods for handling the call user interaction service capability.
Content Based Charging	Package with Web Services containing methods for handling the Contentn based Charging service capability.
Messaging	Package with Web Services containing methods for handling the messaging service capability (SMS, MMS, e-mail).
Messaging User Interaction	Package with Web Services containing methods for handling the message based user interaction service capability (SMS, MMS, e-mail, WAP Push).
Subscriber Profile	Package containing Web Services containing methods for handling the subscriber profile service capabilities.

Package	Description
User Location	Package with Web Services containing methods for handling the user location service capability.
User Status	Package with Web Services containing methods for handling the user status service capability.

Each package, except Access, consists of two or more WSDL files for application initiated operations; one describing the operations applications can invoke on the WebLogic Network Gatekeeper and one describing the operations the WebLogic Network Gatekeeper can invoke on the applications. The first one is denoted the Web Service and the latter the listener interface. Some packages have a third WSDL files, describing the interface for network initiated calls.

Responses to asynchronous requests performed using the Web Service are provided by the listener interface.

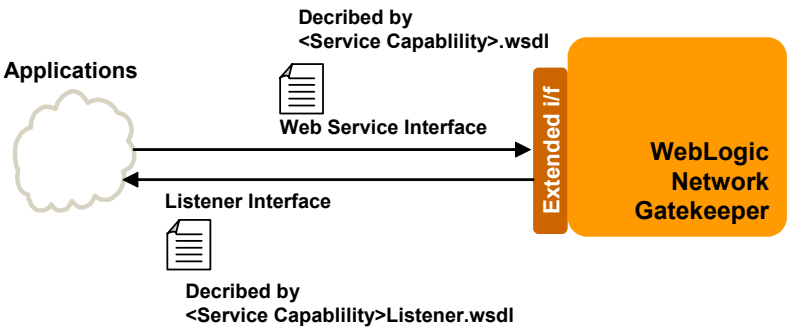

Figure 2-1 Interfaces

The Web Services are located at
http://<URL to WebLogic Network Gatekeeper>/wespa/<service location>/
where <service location> is described in the table below.

The files describing the listener interfaces are located at http://<URL to WebLogic Network Gatekeeper>/wespa/wsdl/<file name>

:

Service Capability	<service location> or <File name>	Describes the
Access	Access	Web Service
	N/A	
	N/A	
Call Control	CallControl	Web Service
	CallListener.wsdl	Listener interface
	NetworkCallListener.wsdl	Network initiated listener interface
Call User Interaction	CallUserInteraction	Web Service
	CallUserInteractionListener.wsdl	Listener interface
	UserInteractionNetworkListener.wsdl	Network initiated listener interface
Content Based Charging	ContentBasedCharging	Web Service
	N/A	Listener interface
	N/A	Network initiated listener interface
Messaging	Messaging	Web Service
	MessagingListener.wsdl	Listener interface
	N/A	Network initiated listener interface
Messaging User Interaction	MessagingUserInteraction	Web Service

Service Capability	<service location> or <File name>	Describes the
	UserInteractionListener.wsdl	Listener interface
	UserInteractionNetworkListener.wsdl	Network initiated listener interface
Subscriber Profile	SubscriberProfile	Web Service
	SubscriberProfileListener.wsdl	Listener interface
	N/A	Network initiated listener interface
User Location	UserLocation	Web Service
	UserLocationListener.wsdl	Listener interface
	N/A	Network initiated listener interface
User Status	UserStatus	Web Service
	UserStatusListener.wsdl	Listener interface
	N/A	Network initiated listener interface
WAP Push	MessagingUserInteraction	Web Service
	UserInteractionListener.wsdl	Listener interface
	N/A	N/A

CHAPTER 3

Access

The following sections provide detailed information about the Access service capability:

Web Service

The access Web Service provides an application with functions for application session management and password management.

When an application session has been created, a loginTicket is returned. This loginTicket is used to identify login session.

applicationLogin

Used by an application to login and retrieve a loginTicket.

Table 3-1 applicationLogin(serviceProvider, application, applicationInstanceGroup, password)

Parameter Name	Type	Description
serviceProvider	xsd:string	ID of the service provider as given by the operator or the service provider.
Input		
application	xsd:string	ID of the application as given by the operator or the service provider.
applicationInstanceGroup	xsd:string	ID of the application instance group as given by the operator or the service provider.
password	xsd:string	Password for the application as given by the operator or the service provider.Note that this may also have been changed by the by the application provider.
Returns		
applicationLoginReturn	xsd:string	Returns a LoginTicket used to identify the login session. This ticket shall be supplied in the SOAP header for each method invocation.
Possible Exceptions		
AccessException		See "AccessException" on page 3-4.
GeneralException		See "GeneralException" on page 3-4.

applicationLogout

Used to login an application to the underlying system. Destroys the login session and the corresponding loginTicket.

Table 3-2 applicationLogout(loginTicket)

Parameter name	Type	Description
Input		
loginTicket	xsd:string	The loginTicket retrieved when logging in.
Returns		
Void.		
Exceptions		
AccessException		See "AccessException" on page 3-4.
GeneralException		See "GeneralException" on page 3-4.

changeApplicationPassword

Used to change the password for an application.

Table 3-3 changeApplicationPassword(loginTicket, oldPassword, newPassword)

Parameter Name	Type	Description
Input		
loginTicket	xsd:string	The loginTicket retrieved when logging in.
oldPassword	xsd:string	The current password.
newPassword	xsd:string	The new password.
Returns		
Void.		

Table 3-3 changeApplicationPassword(loginTicket, oldPassword, newPassword)

Parameter Name	Type	Description
Exceptions		
AccessException		See "AccessException" on page 3-4.
GeneralException		See "GeneralException" on page 3-4.

Exceptions

AccessException

Exceptions of this type are raised when there are error conditions related to the access Web Service. Other error conditions are reported using the exception GeneralException.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

-

Listener interface

There is no listener interface for the Access Web Service.

CHAPTER 4

Call Control

The following sections provide detailed information about the Call Control service capability:

Web Service

The call control Web Service provides an application with functions for call routing, call management, and call leg management. It also provides functions for registering listeners to application initiated and network initiated call sessions.

The service supports establishment of multiparty calls if supported by the underlying platform. Two main usage scenarios are identified; application initiated and network initiated calls.

This section covers application initiated calls.

When a call session has been created, a callTicket is returned. This callTicket is used to identify a given call session and manipulate it.

The following gives an outline of the the Call Control Web Service:

- Create a call
- Add and remove participants in the call
- Hang up all participants and end the call
- Get the originator of the call
- List participants in the call

addListener

Adds a call listener to a call-session initiated by an application. The listener must be a Web Service implemented by the application. The listener receives callbacks with information on the status of the call and error events.

Table 4-1 addListener(callTicket, endpoint)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	Identifier for the call as retrieved when createCall or createEmptyCall was invoked.
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 4-15. An example is `http://www.acompany.com/axis/services/CallListener`

Table 4-1 addListener(callTicket, endpoint)

Parameter Name	Type	Description
Returns		
Void.		
Exceptions		
CallException		See "CallException" on page 4-12.
GeneralException		See "GeneralException" on page 4-12.

addNetworkCallListener

Adds a call listener to call-session initiated from the network. The listener must be a Web Service implemented by the application.

The listener is triggered when:

- the originator address matches the criteria define in the aPartyAddressExpression parameter.
- the destination address matches the criteria define in the bPartyAddressExpression parameter.
- the status of the call matches the criterias given in the eventCriteria parameter.

The listener receives callbacks with information on A-number, B-number and so on in the methods processCallStatusEvent and processCallErrorEvent.

If there are more than one listeners registered for identical criteria, and those listeners are registered from the same application, the listener will be added to a High Availability and Load Balancing list. This means that the network initiated calls matching this criteria will be distributed using a round robin algorithm between the listeners with identical criteria.

If the above is false, and there already is a listener registered for an address that overlaps this listener, an exception is raised.

Table 4-2 addNetworkCallListener(endpoint, aPartyAddressExpression, bPartyAddressExpression, eventCriteria, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 4-15. An example is `http://www.acompany.com/axis/services/CallListener`
aPartyAddressExpression	xsd:string	Originator address in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format.
bPartyAddressExpression	xsd:string	Originator address in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format.
eventCriteria	ArrayOfCallEventCriteria	See "ArrayOfCallEventCriteria" on page 4-12.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
addNetworkCallListenerReturn	xsd:string	Ticket identifying the listener.
Exceptions		
CallException		See "CallException" on page 4-12.
GeneralException		See "GeneralException" on page 4-12.

addParticipant

Adds a new participant to an ongoing call. The application execution continues without waiting for the participant to answer. Notifications when the participant has answered comes in the Web Service defined and registered by the application, see "addListener" on page 4-2.

The ongoing call is identified by the parameter callTicket.

Table 4-3 addParticipant(callTicket, participant, timeout)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	Ticket retrieved when the call was created. See "createCall" on page 4-6 or "createEmptyCall" on page 4-7.
participant	xsd:string	Address to the new participant in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format.
timeout	xsd:int	The time in seconds to wait for an answer from participant.
Returns		
Void		
Exceptions		
CallException		See "CallException" on page 4-12. This exception is raised if the participant could not be added.
GeneralException		See "GeneralException" on page 4-12.

addParticipantWait

Adds a new participant to an ongoing call. The appplication execution holds until the participant answers or until the timeout expires.

The ongoing call is identified by the parameter callTicket.

Table 4-4 addParticipantWait(callTicket, participant, timeout)

Parameter Name	Type	Description
Returns		
callTicket	xsd:string	Ticket retrieved when the call was created. See "createCall" on page 4-6 or "createEmptyCall" on page 4-7.
participant	xsd:string	Address to the new participant in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format.
timeout	xsd:int	The time in seconds to wait for an answer from participant.
Returns		
Void.		
Exceptions		
CallException		See "CallException" on page 4-12. This exception is raised if the participant could not be added.
GeneralException		See "GeneralException" on page 4-12.

createCall

Create a call session with an originator. The originator is identified and the application execution holds until the originator goes off-hook or a timer expires.

Table 4-5 createCall(originator, timeout, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
originator	xsd:string	Address to the originator in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format.

Table 4-5 createCall(originator, timeout, serviceCode, requesterID)

Parameter Name	Type	Description
timeout	xsd:int	The time in seconds to wait for the originator to go off-hook.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
createCallReturn	xsd:string	A CallTicket identifying the call.
Exceptions		
CallException		See "CallException" on page 4-12.
GeneralException		See "GeneralException" on page 4-12.

createEmptyCall

Create a call session without an originator.

Table 4-6 createEmptyCall(serviceCode, requesterID)

Parameter Name	Type	Description
Input		
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
createEmptyCallReturn	xsd:string	A CallTicket identifying the call.
Exceptions		

Table 4-6 createEmptyCall(serviceCode, requesterID)

Parameter Name	Type	Description
CallException		See "CallException" on page 4-12.This exception is raised when the underlying platform filed to create a call object.
GeneralException		See "GeneralException" on page 4-12.

deassign

When a call is deassigned the application will no longer receive events regarding call progress and the callTicket cannot be used for further processing of the call.

The control of the call is handed over to the underlying network.

Table 4-7 deassign(callTicket)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	The CallTicket identifying the call.
Returns		
Void.		
Exceptions		
CallException		See "CallException" on page 4-12. This exception is raised when the call could be deassigned.
GeneralException		See "GeneralException" on page 4-12.

end

Hangs up all the participants and ends the call.

Table 4-8 end(callTicket)

Parameter Name	Type	Description
Returns		
callTicket	xsd:string	The CallTicket identifying the call.
Returns		
Void.		
Exceptions		
CallException		See "CallException" on page 4-12. This exception is raised when the call could not be ended.
GeneralException		See "GeneralException" on page 4-12.

getOriginator

Gets the owner, or originator of the call.

Table 4-9 getOriginator(callTicket)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	The CallTicket identifying the call.
Returns		
getOriginatorReturn	xsd:string	Address of the originator of the call in URI format (tel:<address>). See "Address expressions" on page 4-14 for information on format
Exceptions		

Table 4-9 getOriginator(callTicket)

Parameter Name	Type	Description
CallException		See "CallException" on page 4-12. This exception is raised when the originator could not be retrieved.
GeneralException		See "GeneralException" on page 4-12.

getParticipants

Get a list of all participants in the call. The originator of the call is not included in the list.

Table 4-10 getParticipants(callTicket)

Parameter Name	Type	Description
Retuns		
callTicket	xsd:string	The CallTicket identifying the call.
Returns		
getParticipantsReturn	ArrayofxsdString	See "ArrayOf_xsd_string" on page 4-12. Array of addresses of the participants in the call. See "Address expressions" on page 4-14 for information on format of each individual element in the array.
Exceptions		
CallException		See "CallException" on page 4-12. This exception is raised when the participants could not be retrieved.
GeneralException		See "GeneralException" on page 4-12.

removeNetworkCallListener

Removes a previously registered network call listener. See “addNetworkCallListener” on page 4-3.

Table 4-11 removeNetworkCallListener(listenerTicket)

Parameter Name	Type	Description
Input		
listenerTicket	xsd:string	The ticket retrieved when the network call listener was added.
Returns		
Void.		
Exceptions		
CallException		See “CallException” on page 4-12.This exception is raised when the network call listener could not be removed.
GeneralException		See “GeneralException” on page 4-12.

removeParticipant

Remove an individual participant in an ongoing call session.

Table 4-12 removeParticipant(callTicket, participant)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	A CallTicket identifying the call.
participant	xsd:string	Address of the participant to remove from the call session in URI format (tel:<address>). See “Address expressions” on page 4-14 for information on format.
Returns		
Void.		

Table 4-12 removeParticipant(callTicket, participant)

Parameter Name	Type	Description
Exceptions		
CallException		See "CallException" on page 4-12.This exception is raised when the address of the participant could not be found among the participants in the call.
GeneralException		See "GeneralException" on page 4-12.

Exceptions

CallException

Exceptions of this type are raised when there are error conditions related to the call control Web Service. Other error conditions are reported using the exception GeneralExcepetion.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

ArrayOfCallEventCriteria

Array of call criteria, see "CallEventCriteria" on page 4-13 for information on the individual records in the array.

ArrayOf_xsd_string

Array of strings.

CallErrorCode

Defines errors that can occur during a call-setup or a call.

Enumeration with the following values:

- ERROR_UNKNOWN

- ERROR_NO_ANSWER
- ERROR_BUSY
- ERROR_NO_ROUTE
- ERROR_INVALID_ADDRESS
- ERROR_NO_LISTENER

CallEventCriteria

This data type contains two elements:

- event
- monitorMode

The event part of the CallEventCriteria type defines when and for which call-related events an application shall be notified.

The application can subscribe for notifications related to an originating number and a destination number, either one of them or a combination of both. Notifications can be subscribed for either when:

- An A-party calls a certain B-party.
- An A-party calls any B-party.
- Any A-party calls a certain B-party.
- An A-party goes off hook.

In addition, a set of notifications exists for events that occurs during a call:

- The B-party is busy.
- The B-party is not reachable.
- The B-party goes off-hook
- The B-party hangs up.
- No answer from the B-party.

See "NetworkCallEvent" on page 4-14 for details on valid parameters for the event element.

The monitorMode part of the CallEventCriteria type defines how the application can affect a call. An application that has subscribed for notifications according to the criterias available are notified using either Interrupted mode or Notified mode. Calls that are triggered in Interrupted mode may affect the call, for example by routing the call to another destination or entering another participant in a multiparty conference call. Calls that are triggered in Notfication mode can only be surveilled. See "CallMonitorMode" on page 4-14 for details on valid parameters for the monitorMode element.

CallMonitorMode

Enumeration with one of the following values:

- NOTIFY
- INTERRUPT

NetworkCallEvent

Defines valid call related events that an application can subscribe to during a call:

- ADDRESS_ANALYSED
- ORIGINATING_CALL_AUTHORISED
- TERMINATING_CALL_AUTHORISED
- ORIGINATING_CALL_ATTEMPT
- TERMINATING_CALL_ATTEMPT
- ALERTING
- BUSY
- NO_ANSWER
- CALL_FAILURE
- CALL_ANSWERED
- DISCONNECT

Address expressions

The following rules apply to the address expressions.

Two wildcards are allowed:

- *, which matches zero or more characters.
- ?, which matches exactly one character.

For E.164 addresses (normal telephone numbers), these wildcards are allowed at the beginning or the end of an address.

Examples of valid for E.164 addresses:

- “123” matches specific number.
- “123*” matches all numbers starting with 123 (including 123 itself).
- “123??*” matches all numbers starting with 123 and at least 5 digits long.
- “123???” matches all numbers starting with 123 and exactly 6 digits long.

The following address ranges are illegal:

- “1?3”.
- “1*3”
- “?123*”
- “*”
- “”
- “*123*”
- “*123?”

Legal occurrences of the wildcards should be escaped by a '\' character. For example, to specify a '\' character, '\\' must be used.

Listener interface

The call control listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the call control Web Service, the responses are delivered according to this interface.

callEnded

This method is invoked when the call session has ended. The call session ends when the last participant in the call goes on-hook, or when the application has ended the call. See method "end" on page 4-9.

Table 4-13 callEnded(callTicket)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	A CallTicket identifying the call.
Returns		
Void.		
Exceptions		
-		

deactivate

Used by the underlying system to inform the application that the call session identified by callTicket is no longer valid. The application can not use the call user session no more.

Table 4-14 deactivate(callTicket)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	A CallTicket identifying the call.
Returns		
Void.		
Exceptions		
-		

processCallErrorEvent

Errors related to call sessions are reported using this method.

Table 4-15 processCallErrorEvent(callTicket, error)

Parameter Name	Type	Description
Input		
callTicket	xsd:string.	A CallTicket identifying the call.
error	CallErrorEvent	See "CallErrorEvent" on page 4-18. Type of error.
Returns		
Void.		
Exceptions		
-		

processCallStatusEvent

Table 4-16 processCallStatusEvent(callTicket, event)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	A CallTicket identifying the call.
event	CallStatusEvent	See "CallStatusEvent" on page 4-19. Type of event.
Returns		
Void.		
Exceptions		
-		

Complex data types

CallErrorEvent

Array of call error codes, see "CallErrorCode" on page 4-18 for information on the individual records in the array.

CallErrorCode

Enumeration with one of the following values:

- ERROR_UNKNOWN
- ERROR_NO_ANSWER
- ERROR_BUSY
- ERROR_NO_ROUTE
- ERROR_INVALID_ADDRESS
- ERROR_NO_LISTENER

CallStatusEvent

Array of call status codes, see "CallStatusCode" on page 4-19 for information on the individual records in the array.

CallStatusCode

Enumeration with one of the following values:

- EVENT_CALL_ANSWERED
- EVENT_CALL_HANGUP

Network initiated call listener interface

The network initiated call l listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application.

This listener interface is used for calls that originates from the network. Which phone numbers to survey are defined in the call control Web Service, see "addNetworkCallListener" on page 4-3.

deactivate

Used by the underlying system to inform the application that the call session identified by callTicket is no longer valid. The application can not use the call session no more.

Table 4-17 deactivate(listenerTicket)

Parameter Name	Type	Description
Input		
listnenerTicket	xsd:string	A listenerTicket identifying the call. Retrieved when addNetworkCallListener was invoked.
Returns		
Void.		
Exceptions		
-		

processCall

Call session setup originating from the network, and supervised in monitor mode INTERRUPT, see "addNetworkCallListener" on page 4-3. These call sessions can be controlled by the call control Web Service.

Table 4-18 processCall(listenerTicket, callTicket, originator, participant, event)

Parameter Name	Type	Description
Input		
listenerTicket	xsd:string	A listenerTicket identifying the call. Retrieved when addNetworkCallListener was invoked.
callTicket	xsd:string	A callTicket identifying the call. Used to control the call session using the call control Web Service.
originator	xsd:string	The address of the originator of the call in URI format (tel:<address>).
participant	xsd:string	The address of the destination number dialled by the originator in URI format (tel:<address>).
event	NetworkCallEvent	See "NetworkCallEvent" on page 4-21.
Returns		
Void.		
Exceptions		
-		

processNotification

Call session setup originating from the network, and supervised in monitor mode NOTIFY, see "addNetworkCallListener" on page 4-3. These call sessions can be controlled by the call control Web Service.

Table 4-19 processNotification(listenerTicket, originator, participant, event)

Parameter Name	Type	Description
Input		
listenerTicket	xsd:string	A listenerTicket identifying the call. Retrieved when addNetworkCallListener was invoked.
originator	xsd:string	The address of the originator of the call in URI format (tel:<address>).
participant	xsd:string	The address of the destination number dialled by the originator in URI format (tel:<address>).
event	NetworkCallEvent	See "NetworkCallEvent" on page 4-21.
Returns		
Void.		
Exceptions		
-		

Complex data types

NetworkCallEvent

Enumeration with one of the following values:

- ADDRESS_ANALYSED
- ORIGINATING_CALL_AUTHORISED
- TERMINATING_CALL_AUTHORISED
- ORIGINATING_CALL_ATTEMPT

- TERMINATING_CALL_ATTEMPT
- ALERTING
- BUSY
- NO_ANSWER
- CALL_FAILURE
- CALL_ANSWERED
- DISCONNECT

CHAPTER 5

Call User InterAction

The following sections provide detailed information about the Call User InterAction service capability:

Web Service

The call user interaction Web Service provides an application with functions for callbased user interaction session. This allows an application to perform dialogue handling with one or several end user using IVRs and DTMF as a means of communication.

A call user interaction session depends on an existing call session, which means that a call session must be created before any call user interaction sessions can be created.

When a call user interaction session has been created, a callUiTicket is returned. The callUiTicket is used to identify a call user interaction session.

abortAction

Abort a previously started sendInfo or sendInfoAndCollect request. Invocations of sendInfoWait and sendInfoAndCollectWait can not be aborted using this method.

Table 5-1 abortAction(callUiTicket, assignmentId)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
assignmentId	xsd:int	Identifier of the request to be aborted. The assignmentmentID was returned when sendInfo or sendInfoAndCollect was invoked.
Returns		
Void.		
Exceptions		
CallUIException		See "CallUIException" on page 5-12.
GeneralException		See "GeneralException" on page 5-12.

close

Closes a call user interaction session. The CallUITicket is destroyed and all resources in the network are released.

Table 5-2 close(callUiTicket)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
Returns		
Void.		
Exceptions		
CallUIException		See "CallUIException" on page 5-12.
GeneralException		See "GeneralException" on page 5-12.

createCallUserInteraction

Creates a call user interaction session. Only the session is created, which means that no announcements will be played, nor will no information be collected from the end user until additional methods are invoked.

A call user interaction session uses an ongoing call session, created by the call control Web Service. To connect a call user interaction session with an ongoing call session, the callTicket identifying the call session is used as inparameter when creating a call user interaction session.

Note that some network equipment may only allow CallUserInteraction sessions to be created under certain circumstances, for example with only one participant in the call.

The call user interaction session is identified by the returned callUiTicket.

Table 5-3 createCallUserInteraction(callTicket, participant)

Parameter Name	Type	Description
Input		
callTicket	xsd:string	Ticket that identifies the call session to establish a call user interaction session on.
participant	xsd:string	Address, in URI format (tel:<address>), to the end user, which also must be a participant in the call session, to establish a call user interaction session with. Use "*" to establish call user interaction sessions with all participants in the call. See "Address expressions" on page 4-14 for information on format.
Returns		
createCallUserInteractionReturn	xsd:string	A ticket identifying the call user interaction session. Both this identifier and the ticket identifying the login session, see applicationLogin, (supplied in the security header of the call) must be synchronized in subsequent calls.
Exceptions		
CallUIException		See "CallUIException" on page 5-12. This exception is raised if the participant could not be added.
GeneralException		See "GeneralException" on page 5-12.

sendInfo

Sends an announcement to an end user. The response is reported back to the application using the call user interaction listener interface.

Table 5-4 sendInfo(callUiTicket, endPoint, info, nrOfRepeats, language, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
endPoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 5-13. An example is `http://www.acompany.com/axis/services/CallListener`
info	tsn7:UserInformation	See "UserInformation" on page 5-12 for information on format. The information to send to the IVR. Depends on the IVR equipment used to play announcements. For example, this may be the ID of a prerecorded announcement or a text to be announced.
nrOfRepeats	xsd:int	How many times the announcement message should be repeated.
language	xsd:string	Indicator for language to be used in the announcement. Depends on the IVR equipment used.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendInfoReturn	xsd:int	Id for this unique request in the call user interaction session. Also referred to as assignmentID.
Exceptions		

Table 5-4 sendInfo(callUiTicket, endPoint, info, nrOfRepeats, language, serviceCode, requesterID)

Parameter Name	Type	Description
CallUIException		See "CallUIException" on page 5-12. This exception is raised if the participant could not be added.
GeneralException		See "GeneralException" on page 5-12.

sendInfoAndCollect

Plays an announcement to and collects information from an end user. The application execution continues without waiting for the participant to enter information. The information collected from the user is returned using the Web Service defined and registered by the application, see "sendInfoResult" on page 5-15.

Table 5-5 sendInfoAndCollect(callUiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
endPoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 5-13. An example is `http://www.acompany.com/axis/services/CallUserInteractionListener`
info	tsn7:UserInformation	See "UserInformation" on page 5-12 for information on format. The information to send to the IVR. Depends on the IVR equipment used to play announcements. For example, this may be the ID of a prerecorded announcement or a the text to be announced.

Table 5-5 sendInfoAndCollect(callUiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
minimumLength	xsd:int	The minimum length of input given by the end user for this request to be considered successful. If a timeout or endSequence is encountered before the minimum length is met, it is considered an error.
maximumLength	xsd:int	The maximum length of input allowed for the end user. When maximumLength digits have been collected, the request is considered successful and the result is sent back to the application.
endSequence	xsd:string	The character or characters that will terminate a variable length input (for example a phone number). When this sequence is encountered in the input, and the total number of characters (including end sequence) is greater or equal to minimum length, the request is considered successful and sent back to the application. If the total length is less than minimum length it is considered an error.
startTimeoutSeconds	xsd:int	The first character timeout timer. If no information has been collected from the end user before this timer releases, the input is considered to be faulty.
interCharTimeoutSeconds	xsd:int	The timeout timer between characters collected from the end user. If this timer releases, the input is considered to be faulty if the total length of the input is less than minimum length, otherwise it is considered to be valid.
language	xsd:string	Indicator for language to be used in the announcement. Depends on the IVR equipment used.

Table 5-5 sendInfoAndCollect(callUiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendInfoAndCollectReturn	xsd:int	Id for this unique request in the call user interaction session. Also referred to as assignmentID.
Exceptions		
CallUIException		See "CallUIException" on page 5-12. This exception is raised if the participant could not be added.
GeneralException		See "GeneralException" on page 5-12.

sendInfoAndCollectWait

Plays an announcement to and collects information from an end user. The application execution holds until the information is collected from the end user or a timeout is triggered.

Table 5-6 sendInfoAndCollectWait(callUiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.

Table 5-6 sendInfoAndCollectWait(callUiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
info	tsn7:UserInformation	See "UserInformation" on page 5-12 for information on format. The information to send to the IVR. Depends on the IVR equipment used to play announcements. For example, this may be the ID of a prerecorded announcement or a the text to be announced.
minimumLength	xsd:int	The minimum length of input given by the end user for this request to be considered successful. If a timeout or endSequence is encountered before the minimum length is met, it is considered an error.
maximumLength	xsd:int	The maximum length of input allowed for the end user. When maximumLength digits have been collected, the request is considered successful and the result is sent back to the application.
endSequence	xsd:string	The character or characters that will terminate a variable length input (for example a phone number). When this sequence is encountered in the input, and the total number of characters (including end sequence) is greater or equal to minimum length, the request is considered successful and sent back to the application. If the total length is less than minimum length it is considered an error.
startTimeoutSeconds	xsd:int	The first character timeout timer. If no information has been collected from the end user before this timer releases, the input is considered to be faulty.
interCharTimeoutSeconds	xsd:int	The timeout timer between characters collected from the end user. If this timer releases, the input is considered to be faulty if the total length of the input is less than minimum length, otherwise it is considered to be valid.

Table 5-6 sendInfoAndCollectWait(callUiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
language	xsd:string	Indicator for language to be used in the announcement. Depends on the IVR equipment used.
waitTimeoutSeconds	xsd:int	The time to wait for end user input to complete. If this time expires, a CallUIException with error code RESOURCE_TIMEOUT will be thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendInfoAndCollectWait Return	xsd:string	Input collected from the end user.
Exceptions		
CallUIException		See "CallUIException" on page 5-12.
GeneralException		See "GeneralException" on page 5-12.

sendInfoWait

Plays an announcement to an end user. The application execution holds until the end user the announcement is played to terminates the call, the announcement has been played (including repeats), or a timer expires.

Table 5-7 sendInfoWait(callUiTicket, info, nrOfRepeats, waitTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
info	tsn7:UserInformation	See "UserInformation" on page 5-12 for information on format. The information to send to the IVR. Depends on the IVR equipment used to play announcements. For example, this may be the ID of a prerecorded announcement or a the text to be announced.
nrOfRepeats	xsd:int	The number of times the announcement shall be repeated. Zero (0) indicates that the announcement shall be repeated until the end user terminates the call or the timer expires.
waitTimeoutSeconds	xsd:int	The time to wait for and end user to terminate the call user interaction session. If this time expires, a CallUIException with error code RESOURCE_TIMEOUT will be thrown.
language	xsd:string	Indicator for language to be used in the announcement. Depends on the IVR equipment used.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
Void.		

Table 5-7 sendInfoWait(callUiTicket, info, nrOfRepeats, waitTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
Exceptions		
CallUIException		See "CallUIException" on page 5-12.
GeneralException		See "GeneralException" on page 5-12.

Exceptions

CallUIException

Exceptions of this type are raised when there are error conditions related to the call user interaction Web Service. Other error conditions are reported using the exception GeneralExcepetion.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

UserInformation

Name-value pair, defining the type of user interaction resource to be used. This data is operator defined. Defines which type of information to send to the IVR.

Table 5-8

Name	Type	Description
userInformationType	tns7:UserInformationType	Name part of the name-value pair. See "UserInformationType" on page 5-13.
value	xsd:anyType	Value part of the name-value pair.

UserInformationType

User information types. Defines the type of data defined in UserInformation. Enumeration (xsd:string) with the following values.

Table 5-9

UI_INFO_ID	Use parameter value in datatype UserInformation as an ID of a message defined in an underlying network node. Parameter value in UserInformation shall be xsd:string.
UI_INFO_DATA	Use parameter value in datatype UserInformationData as data to be distributed to an underlying network node.
UI_INFO_ADDRESS	Use parameter value in datatype UserInformation as an address to a message defined in an underlying network node. Parameter value in UserInformation shall be xsd:string.
UI_INFO_BIN_DATA	Use parameter value in datatype UserInformation as binary data to be sent to and underlying network node. Parameter value in UserInformation shall be xsd:string.

UserInformationData

Holder of data to send to an underlying network node.

Table 5-10

infoData	xsd:string	Data to be sent.
infoDataEncodingScheme	xsd:string	Encoding scheme.

Listener interface

The call user interaction listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the call user interaction Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that the call user interaction session identified by callUITicket is no longer valid. The application can not use the call user interaction session no more.

Table 5-11 deactivate(callUiTicket)

Parameter Name	Type	Description
Input		
callUITicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
Returns		
Void.		
Exceptions		
-		

sendInfoError

Errors related to call user interaction sessions are reported using this method.

Table 5-12 sendInfoError(callUiTicket, participant, assignmentId, CallUIErrorCode errorCode)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
participant	xsd:string	The end user for which the error is related.
assignmentId	xsd:int	The ID identifying the request. This ID was retrieved when the application invoked sendInfo or sendInfoAndCollect in the call user interaction Web Service.

Table 5-12 sendInfoError(callUiTicket, participant, assignmentId, CallUIErrorCode errorCode)

Parameter Name	Type	Description
errorCode	CallUIErrorCode	See "CallUIErrorCode" on page 5-16. Type of error.
Returns		
Void.		
Exceptions		
-		

sendInfoResult

Results of call user interaction dialogues initiated by invocation of the asynchronous methods sendInfo and sendInfoAndCollect are reported using this method.

Table 5-13 sendInfoResult(callUiTicket, participant, assignmentId, CallUIResultCode resultCode, info)

Parameter Name	Type	Description
Input		
callUiTicket	xsd:string	Identifier for the call user interaction session retrieved when createCallUserInteraction was invoked.
participant	xsd:string	The end user for which the error is related.
assignmentId	xsd:int	The ID identifying the request. This ID was retrieved when the application invoked sendInfo or sendInfoAndCollect in the call user interaction Web Service.
resultCode	CallUIErrorCode	See "CallUIResultCode" on page 5-16. Type of result.
info	xsd:string	The user input if it was a sendInfoAndCollect request and result code is INFO_COLLECTED, otherwise empty string. The format of this string is determined by the criteria given to the sendInfoAndCollect method.

Table 5-13 sendInfoResult(callUiTicket, participant, assignmentId, CallUIResultCode resultCode, info)

Parameter Name	Type	Description
Returns		
Void.		
Exceptions		
-		

Complex data types

CallErrorEvent

Array of call error codes, see "CallUIErrorCode" on page 5-16 for information on the individual records in the array.

CallUIErrorCode

Enumeration with one of the following values:

- ERROR_UNDEFINED
- ILLEGAL_INFO
- ID_NOT_FOUND
- RESOURCE_UNAVAILABLE
- ILLEGAL_RANGE
- IMPROPER_USER_RESPONSE
- ABANDON
- RESOURCE_TIMEOUT

CallUIResultCode

Enumeration with one of the following values:

- RESULT_UNDEFINED
- INFO_SENT
- INFO_COLLECTED
- NO_INPUT
- TIMEOUT

CHAPTER 6

Content-Based Charging

The following section provides detailed information about the content-based charging service capability.

Web Service

Content based charging makes it possible to charge a subscriber for the content of the service rather than, for example, duration of a call. Charging data records (CDRs) can be generated immediately or in parts. It is also possible to add and reserve amounts to and from accounts.

The Content based charging Web Service provides an application with functions for

- Debit and credit an account
- Debit and credit an account in real time (directly)
- Reserve an amount from an account
- Get amount left of reserved amount

close

Close the content based charging session.

Table 6-1 close(chargingSessionTicket)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
Returns		
-		
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

createChargingSession

Creates a charging session related to the specified user, the charging merchant, and the charging account as well as to the application invoking this method.

Must be invoked before any reservations, credits, or debits can be done.

Table 6-2 createChargingSession(merchantId, accountId, address, corrId, serviceCode, requesterId)

Parameter Name	Type	Description
Input		
merchantId	xsd:string	The charging merchant ID.
accountId	xsd:int	The charging account ID.
address	xsd:string	The address of the subscriber to set data in the profile for. Must be synchronized with the parameter addressPlan.
corrId	impl:CorrelationID	This value can be used to correlate the charging to network activity. See "CorrelationID" on page 6-18.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
createChargingSessionReturn	impl:ChargingSessionID	See "ChargingSessionID" on page 6-18.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

creditAmountWait

Credit (add) an amount of a certain currency to an existing reservation.

Table 6-3 creditAmountWait(chargingSessionTicket, amount, currency, description, requestNumber, releaseAmount)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
amount	xsd:float	The amount to credit.
currency	xsd:string	The currency of the amount to credit. This has to be specified by a three letter combination, according to ISO-4217:1995, "Codes for the representation of currencies and funds".Examples: DEM, EUR, USD.
description	xsd:string	Descriptive text for informational purpose that is,. text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
releaseAmount	xsd:boolean	If set to true, this parameter indicates that the reservation can be freed.
Returns		
creditAmountWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		

Table 6-3 creditAmountWait(chargingSessionTicket, amount, currency, description, requestNumber, releaseAmount)

Parameter Name	Type	Description
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

creditUnitWait

Credit (add) an amount of a certain unit type to an existing reservation.

Table 6-4 creditUnitWait(chargingSessionTicket, volumes, description, requestNumber, releaseUnit)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
volumes	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:	Specifies the number given in the result of the previous operation on this session, or when creating the session.
releaseUnit	xsd:boolean	If set to true, this parameter indicates that the reservation can be freed.
Returns		
creditUnitWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		

Table 6-4 creditUnitWait(chargingSessionTicket, volumes, description, requestNumber, releaseUnit)

Parameter Name	Type	Description
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

debitAmountWait

Debit (withdraw) an amount of a certain currency to an existing reservation.

Table 6-5 debitAmountWait(chargingSessionTicket, amount, currency, description, requestNumber, releaseAmount)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
amount	xsd:float	The amount to debit.
currency	xsd:string	The currency of the amount to credit. This has to be specified by a three letter combination, according to ISO-4217:1995, "Codes for the representation of currencies and funds". Examples: DEM, EUR, USD.
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
releaseAmount	xsd:boolean	If set to true, this parameter indicates that the reservation can be freed.
Returns		

Table 6-5 debitAmountWait(chargingSessionTicket, amount, currency, description, requestNumber, releaseAmount)

Parameter Name	Type	Description
debitAmountWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

debitUnitWait

Debit (withdraw) an amount of a unit type to an existing reservation.

Table 6-6 debitUnitWait(chargingSessionTicket, volumes, description, requestNumber, releaseUnit)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
volumes	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
releaseUnit	xsd:boolean	If set to true, this parameter indicates that the reservation can be freed.

Table 6-6 debitUnitWait(chargingSessionTicket, volumes, description, requestNumber, releaseUnit)

Parameter Name	Type	Description
Returns		
debitUnitWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

directCreditAmountWait

Credit (add) an amount, of a certain currency, directly to a user's account. If a reservation has been made, it will not be influenced.

Table 6-7 directCreditAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
Parameter		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
amount	xsd:float	The amount to credit.
currency	xsd:string	The currency of the amount to credit. This has to be specified by a three letter combination, according to ISO-4217:1995, "Codes for the representation of currencies and funds". Examples: DEM, EUR, USD.
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.

Table 6-7 directCreditAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
directCreditAmountWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

directCreditUnitWait

Credit (add) an amount, of a certain unit type, directly to a user's account. If a reservation has been made, it will not be influenced.

Table 6-8 directCreditUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
volumes	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.

Table 6-8 directCreditUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
directCreditUnitWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

directDebitAmountWait

Debit (withdraw) an amount, of a certain currency, directly from a user's account. If a reservation has been made, it will not be influenced.

Table 6-9 directDebitAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
amount	xsd:float	The amount to debit.
currency	xsd:string	The currency of the amount to debit. This has to be specified by a three letter combination, according to ISO-4217:1995, "Codes for the representation of currencies and funds". Examples: DEM, EUR, USD.

Table 6-9 directDebitAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
directDebitAmountWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

directDebitUnitWait

Debit (withdraw) an amount of a certain unit type directly from a user's account. If a reservation has been made, it will not be influenced.

Table 6-10 directDebitUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
volumes	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.

Table 6-10 directDebitUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
directDebitUnitWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

getAmountLeftWait

Get information on the value of the remaining amount of the reservation.

Table 6-11 getAmountLeftWait(chargingSessionTicket)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
Returns		
getAmountLeftWaitReturn	xsd:float	Amount left in the reservation.
Exception		

Table 6-11 getAmountLeftWait(chargingSessionTicket)

Parameter Name	Type	Description
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

getSessionId

Return a unique ID for this charging session that can be used as a transaction ID in the application when using a reservation.

This can be used, for example, for a final end user purchase confirmation.

Table 6-12 getSessionId(chargingSessionTicket)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
Returns		
getSessionIdReturn	xsd:string	Session ID.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

getUnitLeftWait

Get information on the value of the remaining units of the reservation.

Table 6-13 getUnitLeftWait(chargingSessionTicket)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
Returns		
getUnitLeftWaitReturn	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

rateRequestWait

Request rates for given charging parameters.

Table 6-14 rateRequestWait(chargingSessionTicket, chargingParameters)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
chargingParameters	impl:ArrayOfChargingParameter	See "ArrayOfChargingParameter" on page 6-19.

Table 6-14 rateRequestWait(chargingSessionTicket, chargingParameters)

Parameter Name	Type	Description
Returns		
rateRequestWaitReturn	impl:RateResult	See "RateResult" on page 6-20.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

reserveAmountWait

Reserve an amount, of a certain currency, from a user's account to be used exclusively in this charging session. A reservation has a lifetime for which it is valid.

If invoked more than once, the reservation will be extended to match the total amount in the reservations. This will also re-initialize the reservation lifetime.

Table 6-15 reserveAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
amount	xsd:float	The amount to reserve.
currency	xsd:string	The currency of the amount to reserve. This has to be specified by a three letter combination, according to ISO-4217:1995, "Codes for the representation of currencies and funds". Examples: DEM, EUR, USD.

Table 6-15 reserveAmountWait(chargingSessionTicket, amount, currency, description, requestNumber)

Parameter Name	Type	Description
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
reserveAmountWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

reserveUnitWait

Reserve an amount, of a certain unit type, from a user's account to be used exclusively in this charging session. A reservation has a lifetime for which it is valid.

If invoked more than once, the reservation will be extended to match the total amount in the reservations. This will also re-initialize the reservation lifetime.

Table 6-16 reserveUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
Input		
chargingSessionTicket	xsd:string	ID of the content based charging session, retrieved when it was created. See "createChargingSession" on page 6-3.
volumes	impl:ArrayOfVolume	See "ArrayOfVolume" on page 6-18.

Table 6-16 reserveUnitWait(chargingSessionTicket, volumes, description, requestNumber)

Parameter Name	Type	Description
description	xsd:string	Descriptive text for informational purpose. That is, text presented on the bill and used in communication towards the user.
requestNumber	xsd:int	Specifies the number given in the result of the previous operation on this session, or when creating the session.
Returns		
reserveUnitWaitReturn	xsd:int	Request number. This number shall be used as inparameter in the next request in the charging session.
Exception		
ContentBasedChargingException		See "ContentBasedChargingException" on page 6-17.
GeneralException		See "GeneralException" on page 6-17.

Exceptions

ContentBasedChargingException

Exception of this type are raised when there are error conditions related to the Content Based Charging Web Service. Other error conditions are reported using GeneralException.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

CorrelationID

This value can be used to correlate the charging to network activity.

Table 6-17

Name	Type	Description
correlation	xsd:int	ID of the correlation.
corrType	xsd:int	Type of correlation.

ChargingSessionID

Holder for chargingTicket and initial request number.

Table 6-18

Name	Type	Description
chargingTicket	xsd:string	Identifier for the charging session.
initialRequestNumber	xsd:int	Request number to be used as inparameter in the first charging operation.

ArrayOfVolume

Table 6-19

Name	Type	Description
Array of Volume		See "Volume" on page 6-19.

Volume

Holder for amount and unit. Either one is used.

Table 6-20

Name	Type	Description
amount	xsd:float	Amount to be used.
unit	xsd:int	Number representing the units to be used. • 0 means undefined • 1 means number • 2 means octets • 3 means seconds • 4 means minutes • 5 means hours • 6 means days

ArrayOfChargingParameter

Table 6-21

Name	Type	Description
ArrayofChargingParameter		See "ChargingParameter" on page 6-19.

ChargingParameter

Holder for the value.

Table 6-22

Name	Type	Description
parameterId	xsd:int	ID of the parameter.
chargingPValue	impl:ChargingParameterValue	See "ChargingParameterValue" on page 6-20.

ChargingParameterValue

Defines datatype and value of the parameter. Either one is used.

Table 6-23

Name	Type	Description
intValue	xsd:int	Value given as an int.
floatValue	xsd:float	Value given as a float.
booleanValue	xsd:boolean	Value given as a boolean.
stringValue	xsd:string	Value given as a string.

RateResult

Holder of results from rating requests.

Table 6-24

Name	Type	Description
rates	impl:PriceVolume	See "PriceVolume" on page 6-20.
validityTimeLeftMs	xsd:int	Defines for how long the rate is valid. Given in milliseconds.

PriceVolume

Defines datatype and value of the parameter. If amount is used, currency is also used.

Table 6-25

Name	Type	Value
amount	xsd:float	The amount.
currency	xsd:string	The currency used in the parameter amount.
unitVolume	impl:Volume	See "Volume" on page 6-19.

CHAPTER 7

Messaging

The following sections provide detailed information about the Messaging service capability:

Web Service

The messaging Web Service provides an application with functions for sending and receiving SMS, MMS, and e-mail messages.

closeMailBox

Closes a previously opened mailbox.

Table 7-1 closeMailBox(mailboxTicket)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked. See "openMailbox" on page 7-9.
Returns		
Void.		
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

deleteMessages

Deletes one or more messages from a mailbox folder.

Table 7-2 deleteMessages(mailboxTicket, folder, messageID, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked. See "openMailbox" on page 7-9.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.

Table 7-2 deleteMessages(mailboxTicket, folder, messageID, requesterID)

Parameter Name	Type	Description
messageID	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 7-13.
requesterID	xsd:string	Application ID as provided by the operator.
Returns		
Void.		
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

disableMessagingNotification

Disables a previously subscription for notifications on events related to a mailbox.

Table 7-3 disableMessagingNotification(notificationTicket)

Parameter Name	Type	Description
Input		
notificationTicket	xsd:string	Identifier for the notification listener as retrieved when enableMessagingNotification was invoked. See "enableMessagingNotification" on page 7-4.
Returns		
Void.		
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

enableMessagingNotification

Subscribes for notifications on events related to a mailbox.

Table 7-4 enableMessagingNotification(endPoint, mailbox, mailboxPassword, notificationCriteria, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 7-21. An example is `http://www.acompany.com/axis/services/MessagingListener`
mailbox	xsd:string	The ID of the mailbox to subscribe for notifications on.
mailboxPassword	xsd:string	The password associated with the mailbox.
notificationCriteria	impl:NotificationCriteria	See "NotificationCriteria" on page 7-21. Defines which events to subscribe for notifications on.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
enableMessagingNotificationReturn	xsd:string	ID of the request. This ID is supplied in newMessageAvailable to keep track of the different request-response pairs.
Exceptions		

Table 7-4 enableMessagingNotification(endPoint, mailbox, mailboxPassword, notificationCriteria, serviceCode, requesterID)

Parameter Name	Type	Description
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

getMMS

Gets an MMS from the mailbox. The message is returned as an SOAP attachment in mime format.

Table 7-5 getMMS(mailboxTicket, folder, messageID, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	ID of the mailbox as retrieved when openMailbox was invoked. See "openMailbox" on page 7-9.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.
messageID	xsd:string	ID of the message as given in newMessageAvailable.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
Void		
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

getMessageProperties

Gets properties of a message. Properties includes originator address, destination address, subject, and so on.

Table 7-6 getMessageProperties(mailboxTicket, folder, messageID, propertyTypes, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.
messageID	xsd:string	ID of the message as given in newMessageAvailable.
propertyTypes	impl:ArrayOfMessagingPropertyName	See "ArrayOfMessagingPropertyName" on page 7-13. Defines which properties to get.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getMessagePropertiesReturn		impl:ArrayOfMessagingProperty. See "ArrayOfMessagingProperty" on page 7-13. Properties for the message.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

getSMS

Fetches an SMS from the mailbox.

Table 7-7 getSMS(mailboxTicket, folder, messageID, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.
messageID	xsd:string	ID of the message as given in newMessageAvailable.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getSMSReturn	xsd:string	The message text.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

listMessages

List messages in a mailbox folder.

Table 7-8 listMessages(mailboxTicket, folder, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.

Table 7-8 listMessages(mailboxTicket, folder, requesterID)

Parameter Name	Type	Description
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
listMessagesReturn	impl:ArrayOfMessageDescription	See "ArrayOfMessageDescription" on page 7-13. An array of message descriptions.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

listNewMessages

List unread messages in a mailbox folder.

Table 7-9 listNewMessages(mailboxTicket, folder, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the message is stored in.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 7-9 listNewMessages(mailboxTicket, folder, requesterID)

Parameter Name	Type	Description
listNewMessagesReturn	impl:ArrayOfMessageDescription	See "ArrayOfMessageDescription" on page 7-13. An array of message descriptions.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

openMailbox

Opens a mailbox. Returns a mailboxTicket to be used in messaging operations. The mailbox name and its associated password is provided by the operator.

Table 7-10 openMailbox(mailbox, password, requesterID)

Parameter Name	Type	Description
Input		
mailbox	xsd:string	Name of the mailbox.
password	xsd:string	Password for the mailbox.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
openMailboxReturn	xsd:string	The mailboxTicket used in operations towards the mailbox.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

sendEMail

Sends an e-mail to one or more destination addresses. The message shall be provided as a SOAP attachment in mime format.

Table 7-11 sendEMail(mailboxTicket, messageProperties, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
messageProperties	impl:ArrayOfMessagingProperty	See "ArrayOfMessagingProperty" on page 7-13. Properties for the message. For example destination address and subject.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendEMailReturn	impl:ArrayOfMessageSendResult	See "ArrayOfMessageSendResult" on page 7-13. Contains information on identifier for the message, status of the message and so on for each destination address.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

sendMMS

Sends an MMS to one or more destination addresses. The message shall be provided as a SOAP attachment in mime format.

Table 7-12 sendMMS(mailboxTicket, messageProperties, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
messageProperties	impl:ArrayOfMessagingProperty	See "ArrayOfMessagingProperty" on page 7-13. Properties for the message. For example destination address and subject.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendMMSReturn	impl:ArrayOfMessageSendResult	See "ArrayOfMessageSendResult" on page 7-13. Contains information on identifier for the message, status of the message and so on for each destination address.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

sendSMS

Sends an SMS to one or more destination addresses. Message delivery status can be checked by reading the properties for the message IDs returned by this method.

Table 7-13 sendSMS(mailboxTicket, message, messageProperties, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
mailboxTicket	xsd:string	Identifier for the mailbox as retrieved when openMailbox was invoked.
messageProperties	impl:ArrayOfMessagingProperty	See "ArrayOfMessagingProperty" on page 7-13. Properties for the message. For example destination address.
message	xsd:string	The text of the message.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendSMSReturn	impl:ArrayOfMessageSendResult	See "ArrayOfMessageSendResult" on page 7-13. Contains information on identifier for the message, status of the message and so on for each destination address.
Exceptions		
MessagingException		See "MessagingException" on page 7-13.
GeneralException		See "GeneralException" on page 7-13.

Exceptions

MessagingException

Exception of this type are raised when there are error conditions related to the messaging Web Service. Other error conditions are reported using the exception GeneralExcepetion.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

ArrayOfMessageDescription

Array of message descriptions. See "MessageDescription" on page 7-15 for information on the individual records in the array.

ArrayOfMessageSendResult

Array of results of a result of a message send operation. See "MessageSendResult" on page 7-16 for information on the individual records in the array.

ArrayOfMessagingProperty

Array of message properties. See "MessagingProperty" on page 7-18 for information on the individual records in the array.

ArrayOfMessagingPropertyName

Array of message property names. See "MessagingPropertyName" on page 7-18 for information on the individual records in the array.

ArrayOf_xsd_string

Array of strings.

ContentType

Content type defined for MMS messages. Enumeration (xsd:string) with the following values:

Table 7-14

Value	Description
CT_UNDEFINED	Undefined content type.
CT_APPLICATION_MULTIPART_MIXED	MIME multiparty/mixed.
CT_APPLICATION_MULTIPART_RELATED	MIME multiparty/related.
CT_APPLICATION_SMIL	SMIL
CT_IMAGE_GIF	GIF image.
CT_IMAGE_JPEG	JPEG image
CT_IMAGE_PNG	PNG image.
CT_IMAGE_TIFF	TIFF image.
CT_IMAGE_WBMP	WAP BMP image.
CT_TEXT_HTML	HTML.
CT_TEXT_PLAIN	Plain text.
CT_TEXT_WML	WAP Mark-up language.
CT_AUDIO_AMR	AMR audio.
CT_AUDIO_WAVE	Wave audio.
CT_AUDIO_MP3	MP3 audio.
CT_AUDIO_AU	AU audio.
CT_AUDIO_AIF	AIF audio.
CT_AUDIO_SND	SND audio.
CT_AUDIO_RA	RA audio.
CT_AUDIO_MID	MIDI audio.

MailboxFolder

Defines the folders in a mailbox. Enumeration (xsd:string) with the following values.

Table 7-15

Value	Description
MESSAGE_MAILBOX_UNDEFINED	Undefined mailbox folder.
MESSAGE_MAILBOX_OUTBOX	Outbox folder.
MESSAGE_MAILBOX_INBOX	Inbox folder.
MESSAGE_MAILBOX_DRAFTS	Draft folder.
MESSAGE_MAILBOX_TRASH	Trash folder.
MESSAGE_MAILBOX_TEMPLATES	Templates folder.

MessageDescription

Contains the ID for a message and the type of message.

Table 7-16

Name	Type	Description
messageId	xsd:string	ID of the message.
format	intf:MessageFormatType	See "MessageFormatType" on page 7-15,

MessageFormatType

Information on the format of a message. Enumeration (xd:string) with one of the following values.

Table 7-17

Value	Description
MESSAGE_FORMAT_AU	Audio message.
MESSAGE_FORMAT_BINARY	Binary message.
MESSAGE_FORMAT_MIME	MIME message.
MESSAGE_FORMAT_MM	Multimedia message.
MESSAGE_FORMAT_TEXT	Text message.
MESSAGE_FORMAT_UNDEFINED	Undefined.
MESSAGE_FORMAT_UUENCODED	UU-encoded message.
MESSAGE_FORMAT_WAVE	Wave audio message.

MessageSendResult

Information about the result of a send message operation for an individual destination address.

Table 7-18

Name	Type	Description
address	xsd:string	Destination address, in URI format (tel:<address>), (MSISDN or e-mail address).
messageID	xsd:string	ID of the message.
sendStatus	intf:MessageSendStatus	See "MessageSendStatus" on page 7-17.

MessageSendStatus

Defines the status of a message. Enumeration (xsd:string) with one of the following values.

Table 7-19

Value	Description
MESSAGE_SEND_OK	The message was successfully sent.
MESSAGE_SEND_ERROR	An error occurred when the message was sent.

MessageStatusType

Definition of statuses a message can have. Enumeration (xsd:string) with the following values.

Table 7-20

Value	Description
MESSAGE_STATUS_UNDEFINED	Undefined.
MESSAGE_STATUS_READ_MESSAGE	Read.
MESSAGE_STATUS_UNREAD_MESSAGE	Unread.
MESSAGE_STATUS_FORWARDED_MESSAGE	Has been forwarded to another address.
MESSAGE_STATUS_REPLIED_TO_MESSAGE	Has been replied to.
MESSAGE_STATUS_SAVED_OR_UNSENT_MESSAGE	Saved or not yet sent.
MESSAGE_STATUS_NOTIFICATION_THAT_A_MESSAGE_WAS_DELIVERED	Delivered to destination address(es).
MESSAGE_STATUS_NOTIFICATION_THAT_A_MESSAGE_WAS_READ	Destination addressee has read the message.
MESSAGE_STATUS_NOTIFICATION_THAT_A_MESSAGE_WAS_NOT_DELIVERED	Message was not delivered to the destination.
MESSAGE_STATUS_NOTIFICATION_THAT_A_MESSAGE_WAS_NOT_READ	Message is not read.

MessagingException

Table 7-21

Name	Value	Description
exceptionMessage	xsd:string	Error message in text-format.
errorCode	xsd:int	Error code.

MessagingProperty

Table 7-22

Name	Type	Description
messagingPropertyName	intf:MessagingPropertyName	See "MessagingPropertyName" on page 7-18.
value	xsd:anyType	See "MessagingPropertyName" on page 7-18 for which data type that corresponds to which MessagingPropertyname.

MessagingPropertyName

Properties of a message. Enumeration (xsd:string) with the following values.

Table 7-23

Value	Description
MESSAGE_UNDEFINED	Undefined. Null.
MESSAGE_ID	ID of the message. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_SUBJECT	Subject of the message. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_DATE_SENT	Date when the message was sent. Use xsd:string in xsd:anyType in MessagingProperty.

Table 7-23

Value	Description
MESSAGE_DATE_RECEIVED	Date the message was received. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_DATE_CHANGED	Date the message was changed. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_SENT_FROM	Originator address. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_SENT_TO	Destination address. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_CC_TO	Carbon copy address. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_BCC_TO	Blind carbon copy address. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_SIZE	Size of the message. Use xsd:int in xsd:anyType in MessagingProperty.
MESSAGE_PRIORITY	Priority the message was sent with. Use Priority in xsd:anyType in MessagingProperty.
MESSAGE_FORMAT	Format of the message. Use MessageFormatType in xsd:anyType in MessagingProperty.
MESSAGE_STATUS	Status of the message. Use MessageStatusType in xsd:anyType in MessagingProperty.
MESSAGE_VALIDITY_PERIOD	Validity period of the message. Use xsd:string in xsd:anyType in MessagingProperty.
MESSAGE_ENCODING_TYPE	Encoding type of the message. Use xsd:string in xsd:anyType in MessagingProperty.

Priority

Priority of a message. Enumeration (xsd:string) with one of the following values.

Table 7-24

Value	Description
LOW	Low priority.
HIGH	High priority.
NORMAL	Normal priority.

MmsContent

Defines the content of an individual part of an MMS.

Table 7-25

Name	Type	Description
data	xsd:byte	Binary part of an element in an MMS message.
type	intf:ContentType	See "ContentType" on page 7-14.The type of content in the data field.
contentID	xsd:string	ID of the content.

MmsMessage

Container for content and properties for an MMS.

Table 7-26

Name	Type	Description
content	intf:MmsContent	See "MmsContent" on page 7-20.
properties	intf:MessagingProperty	See "MessagingProperty" on page 7-18.

NotificationCriteria

Defines for which events notifications shall be sent to a listener.

Enumeration (xsd:string) with one of the values.

Table 7-27

Value	Description
NC_NEW_MESSAGE_ARRIVED	Send notification when a new message arrives to the inbox folder.
NC_MESSAGE_DELIVERED_ACK	Send notification when a delivery acknowledgment has arrived.

Listener interface

The messaging listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the messaging Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that the session identified by notificationTicket is no longer valid. The application can not use the session no more.

Table 7-28 deactivate(notificationTicket)

Parameter Name	Type	Description
Input		
notificationTicket	xsd:string	ID of the session. The ID was returned when MessagingProperty was invoked.
Returns		
Void.		
Exceptions		
-		

messageDeliveryAck

Message delivery receipts for messages are reported using this method.

Table 7-29 messageDeliveryAck(notificationTicket, messageId, messageStatus)

Input		
notificationTicket	xsd:string	A ticket identifying the notification session.
messageId	xsd:string	ID of the message.
messageStatus	impl:MessageStatusType	See "MessageStatusType" on page 7-17.
Returns		
Void.		
Exceptions		
-		

newMessageAvailable

Information on arrival of new messages is reported using this method.

Table 7-30 newMessageAvailable(notificationTicket, mailbox, folder, messageDescr)

Parameter Name	Type	Description
Input		
notificationTicket	xsd:string	ID of the session. This ticket was given when enableMessagingNotification was invoked.
mailbox	xsd:string	ID of the mailbox that holds a new message.
folder	impl:MailboxFolder	See "MailboxFolder" on page 7-15. The folder the new message is stored in.

Table 7-30 newMessageAvailable(notificationTicket, mailbox, folder, messageDescr)

Parameter Name	Type	Description
messageDescr	impl:MessageDescription	See "MessageDescription" on page 7-15. Description of the message.
Returns		
Void.		
Exceptions		
-		

Complex data types

A subset of the ones described in "Complex data types" on page 7-13.

CHAPTER 8

Messaging User Interaction

The following sections provide detailed information about the Messaging User Interaction service capability:

Web Service

The Messaging User interaction Web Service provides an application with functions to interact with end users. The application interacts with end users through text messages, for example SMS or USSD messages.

SMS based user interaction provides application initiated SMSes with a transaction ID to connect requesting/prompting SMSes with end user's replies.

USSD messages from an application can be purely informative or they can prompt the end user to reply. USSD messages can also be used by the end user to initiate service sessions with applications. When initiating service sessions or replying to an application generated USSD message, the end user can only use the terminal's key set (0-9, *, #). The application can use any type of character supported by the end user's terminal.

createUI

Creates a user interaction session with a terminal.

Table 8-1 createUI(participant)

Parameter Name	Type	Description
Input		
participant	xsd:string	Address of the telephony terminal in URI format (tel:<address>).
Returns		
uiSessionTicket	xsd:int	Identifier for the session. Both this identifier and the ticket identifying the login session, see applicationLogin, (supplied in the security header of the call) must be synchronized in subsequent calls.
Exceptions		
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

addNetworkUIListener

Subscribes for notifications on events related terminal initiated user interaction sessions.

The combination of a matching aPartyAddressExpression, bPartyAddressExpression and userInteractionCode triggers the Web Service located at endpoint.

Table 8-2 addNetworkUIListener(endPoint, aPartyAddressExpression, bPartyAddressExpression, userInteractionCode, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 8-14. An example is `http://www.acompany.com:8080/axis/services/UserInteractionListener`
aPartyAddressExpression	xsd:string	The originating address or address range for which the notification is requested in URI format (tel:<address>). Wildcards, like * and ? are supported.
bPartyAddressExpression	xsd:string	Destination address or address range for which the notification is requested in URI format (tel:<address>). Only useful in cases when the destination is other than the application, for example when listening to message based user interaction sessions.
userInteractionCode	xsd:string	A 2-digit code indicating the UI to be triggered. Defined by the operator.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
addNetworkUIListenerReturn	xsd:string	An ID for the listener. This ID is used when disabling the notification. See "removeNetworkUIListener" on page 8-5.

Table 8-2 addNetworkUIListener(endPoint, aPartyAddressExpression, bPartyAddressExpression, userInteractionCode, serviceCode, requesterID)

Parameter Name	Type	Description
Exceptions		
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

closeUI

Terminates a user interaction session.

Table 8-3 closeUI(uiTicket)

Parameter Name	Type	Description
Input		
uiSessionTicket	xsd:string	ID of the user interaction session, retrieved when it was created. See "createUI" on page 8-2.
Returns		
Void		
Exceptions		
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

removeNetworkUIListener

Removes a previously registered network triggered user interaction listener.

Table 8-4 removeNetworkUIListener(notificationTicket, requesterID)

Parameter Name	Type	Description
Input		
listenerID	xsd:string	ID of the listener, retrieved when it was registered. See "addNetworkUIListener" on page 8-3.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
Void		
Exceptions		
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

sendInfo

Send predefined information via SMS or USSD to the telephony terminal. Asynchronous method. The result will be notified on the user interaction listener.

Table 8-5 sendInfo(uiTicket, endPoint, info, noRepeats, language, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
uiTicket	xsd:string	Ticket identifying the user interaction session. The ID was retrieved when createUI or processUINotification was invoked.
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 8-14. An example is `http://www.acompany.com/axis/services/MessagingListener`

Table 8-5 sendInfo(uiTicket, endPoint, info, noRepeats, language, serviceCode, requesterID)

Parameter Name	Type	Description
info	impl:UserInformation	Defines the type of user interaction resource to be used. See "UserInformation" on page 8-12.
noRepeats	xsd:int	The number of times the message, defined in the parameter info, shall be repeated. For message based user interaction sessions, this parameter is ignored.
language	xsd:string	Language code for the message. Defined by the operator.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendInfoReturn	xsd:int	Ticket to identify the request-response pairs within the user interaction session.
Exceptions		
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

sendInfoWait

Send predefined information via SMS or USSD to the telephony terminal. Synchronous method.

Table 8-6 sendInfoWait(uiTicket, info, noRepeats, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
uiTicket	xsd:string	Ticket identifying the user interaction session. The ID was retrieved when createUI or processUINotification was invoked.
info	impl:UserInformation	Defines the type of user interaction resource to be used. See "UserInformation" on page 8-12.
noRepeats	xsd:int	The number of times the message, defined in the parameter info, shall be repeated. For message based user interaction sessions, this parameter is ignored.
language	xsd:string	Language code for he message. Defined by the operator.
waitTimeoutSeconds	xsd:int	The time to wait for the message to be delivered to the telephony terminal. An UIException exception occurs if this time is overdue.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
void		
Exceptions		

Table 8-6 sendInfoWait(uiTicket, info, noRepeats, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
UIException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

sendInfoAndCollect

Send predefined information to the telephony terminal via SMS or USSD and collect user input from the end user via SMS or USSD. Asynchronous method. The result, containing user input, will be notified on the user interaction listener.

Table 8-7 sendInfoAndCollect(uiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
uiTicket	xsd:string	Ticket identifying the user interaction session. The ID was retrieved when createUI or processUINotification was invoked.
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 8-14. An example is `http://www.acompany.com/axis/services/MessagingListener`
info	impl:UserInformation	Defines the type of user interaction resource to be used. See "UserInformation" on page 8-12.
minimumLength	xsd:int	The minimum length of input for this request to be considered successful. If a timeout or endSequence is encountered before the minimum length is met, it is considered an error. For message based user interaction sessions, this parameter is ignored.

Table 8-7 sendInfoAndCollect(uiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
maximumLength	xsd:int	The maximum length of input. When maximumLength digits have been collected, the request is considered successful, and sent back to the application. For message based user interaction sessions, this parameter is ignored.
endSequence	xsd:string	The character or characters that will terminate a variable length input. When this sequence is encountered in the input, and the total number of characters (including end sequence) is greater or equal to minimum length, the request is considered successful and sent back to the application. If the total length is less than minimum length it is considered an error. For message based user interaction sessions, this parameter is ignored.
startTimeoutSeconds	xsd:int	The first character timeout timer. If no character has been input before this timer releases, the input is considered an error. For message based user interaction sessions, this parameter is ignored.
interCharTimeoutSeconds	xsd:int	The timeout timer between characters. If this timer releases, the input is considered an error if the total length of the input is less than minimum length, otherwise a success. For message based user interaction sessions, this parameter is ignored.
language	xsd:string	Language code for the message. Defined by the operator.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 8-7 sendInfoAndCollect(uiTicket, endPoint, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, serviceCode, requesterID)

Parameter Name	Type	Description
sendInfoAndCollectReturn	xsd:int	Ticket to identify the request-response pairs.
Exceptions		
MessagingException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

sendInfoAndCollectWait

Send predefined information to the telephony terminal via SMS or USSD and collect user input from the end user via SMS or USSD. Synchronous method.

Table 8-8 sendInfoAndCollectWait(uiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
uiTicket	xsd:string	Ticket identifying the user interaction session. The ID was retrieved when createUI or processUINotification was invoked.
info	impl:UserInformation	Defines the type of user interaction resource to be used. See "UserInformation" on page 8-12.
minimumLength	xsd:int	The minimum length of input, for this request to be considered successful. If a timeout or endSequence is encountered before the minimum length is met, it is considered an error. For message based user interaction sessions, this parameter is ignored.
maximumLength	xsd:int	The maximum length of input. When maximumLength digits have been collected, the request is considered successful, and sent back to the application.For message based user interaction sessions, this parameter is ignored.

Table 8-8 sendInfoAndCollectWait(uiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
endSequence	xsd:string	The character or characters that will terminate a variable length input. When this sequence is encountered in the input, and the total number of characters (including end sequence) is greater or equal to minimum length, the request is considered successful and sent back to the application. If the total length is less than minimum length it is considered an error. For message based user interaction sessions, this parameter is ignored.
startTimeoutSeconds	xsd:int	The first character timeout timer. If no character has been input before this timer releases, the input is considered an error. For message based user interaction sessions, this parameter is ignored.
interCharTimeoutSeconds	xsd:int	The timeout timer between characters. If this timer releases, the input is considered an error if the total length of the input is less than minimum length, otherwise a success. For message based user interaction sessions, this parameter is ignored.
language	xsd:string	Language code for he message. Defined by the operator.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
sendInfoAndCollectWait Return	xsd:string	Collected end user input.The format of this string is determined by the criteria given to the sendInfoAndCollect method.
Exceptions		

Table 8-8 sendInfoAndCollectWait(uiTicket, info, minimumLength, maximumLength, endSequence, startTimeoutSeconds, interCharTimeoutSeconds, language, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
MessagingException		See "UIException" on page 8-12.
GeneralException		See "GeneralException" on page 8-12.

Exceptions

UIException

Exception of this type are raised when there are error conditions related to the Messaging User Interaction Web Service. Other error conditions are reported using GeneralException.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

UserInformation

Name-value pair, defining the type of user interaction resource to be used. This data is operator defined.

Table 8-9

Name	Type	Description
userInformationType	tns7:UserInformationType	Name part of the name-value pair. See "UserInformationType" on page 8-13.
value	xsd:anyType	Value part of the name-value pair.

UserInformationType

User information types. Defines the type of data defined in UserInformation. Enumeration (xsd:string) with the following values.

Table 8-10

Value	Description
UI_INFO_ID	Use parameter value in datatype UserInformation as an ID of a message defined in an underlying network node. Parameter value in UserInformation shall be xsd:string.
UI_INFO_DATA	Use parameter value in datatype UserInformationData as data to be distributed to an underlying network node.
UI_INFO_ADDRESS	Use parameter value in datatype UserInformation as an address to a message defined in an underlying network node. Parameter value in UserInformation shall be xsd:string.
UI_INFO_BIN_DATA	Use parameter value in datatype UserInformation as binary data to be sent to and underlying network node. Parameter value in UserInformation shall be xsd:string.

UserInformationData

Holder of data to send to an underlying network node.

Table 8-11

Name	Type	Description
infoData	xsd:string	Data to be sent.
infoDataEncodingScheme	xsd:string	For message based user interaction sessions, this parameter is ignored, however it must not be null.

UIEventDataTypeCode

Defines the type of data retrieved. Enumeration (xsd:string) with the following values.

Table 8-12

Value	Description
EVENT_DATA_TYPE_UNDEFINED	The format of the data is undefined.
EVENT_DATA_TYPE_UNSPECIFIED	The format of the data is unspecified.
EVENT_DATA_TYPE_TEXT	The data is in text format.

Listener interface

The Messaging User Interaction listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the Messaging User Interaction Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that the session identified by uiSessionTicket is no longer valid. The application can not use the session no more.

Table 8-13 deactivate(assignmentId)

Parameter Name	Type	Description
Input		
assignmentId	xsd:string	ID of the session. The ID was returned when createUI was invoked.
Returns		
Void.		
Exceptions		
-		

sendInfoResult

Successful responses to invocations of asynchronous are reported using this method.

Table 8-14 sendInfoResult(assignmentId, resultCode, collectedInfo)

Parameter Name	Type	Description
Input		
assignmentId	xsd:int	ID of the session. The ID was returned when createUI was invoked.
resultCode	impl:UIResultCode	Defines the result of an invocation of the method sendInfo (page 49) or sendInfoAndCollect (page 52). See "UIResultCode" on page 8-16.
collectedInfo	xsd:string	The user input if it was a sendInfoAndCollect request and result code is INFO_COLLECTED (see "UIResultCode" on page 8-16), otherwise empty string. The format of the string is determined by the criteria given to in sendInfoAndCollect or sendInfoAndCollectWait.
Returns		
Void.		
Exceptions		
-		

sendInfoError

Failed responses to invocations of asynchronous are reported using this method.

Table 8-15 sendInfoError(assignmentId, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
assignmentId	xsd:int	ID of the session. The ID was returned when createUI was invoked.
errorCode	xsd:string	Error code.
errorMsg	xsd:string	Textual error description.
Returns		
Void.		
Exceptions		
-		

Complex data types

UIResultCode

Defines the result of an invocation of the method sendInfo or sendInfoAndCollect.

Enumeration (xsd:string) with the following values.

Table 8-16

Value	Description
RESULT_UNDEFINED	The result of the operation is unknown.
INFO_SENT	Information sent to the telephony terminal.
INFO_COLLECTED	Information collected from the user of the telephony terminal.

Table 8-16

Value	Description
NO_INPUT	No input collected from the user of the telephony terminal.
TIMEOUT	The request to collect information from the user of the telephony terminal timed out. The timeout value is defined by the operator.

Network Listener interface

The Messaging User Interaction Network Listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application.

In contrast to the Listener interface it listens to user interaction sessions initiated from the network, an ultimately from an telephony terminal.

That is, When an application performs asynchronous requests from the Messaging User Interaction Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that the session identified by uiSessionTicket is no longer valid. The application can not use the session no more.

Table 8-17 deactivate(notificationTicket)

Parameter Name	Type	Description
Input		
assignmentId	xsd:string	ID of the session. The notification registration that caused this notification. The ID was returned when createUI was invoked.
Returns		
Void.		
Exceptions		
-		

processUINotification

Successful responses to invocations of asynchronous are reported using this method.

Table 8-18 processUINotification(notificationTicket, uiTicket, originator, destination, userInteractionCode, dataTypeCode, dataString)

Parameter Name	Type	Description
Input		
notificationTicket	xsd:string	ID of the session. The ID was returned when addNetworkUIListener was invoked.
uiTicket	xsd:string	ID of the session.
originator	xsd:string	Address, in URI format (tel:<address>), of the telephony terminal that initiated the user interaction session.
destination	xsd:string	Address, in URI format (tel:<address>), of the application or service.
userInteractionCode	xsd:string	A 2-digit code indicating the UI that was triggered. Defined by the operator.
dataTypeCode	impl:UIEventDataTypeCode	Defines the type of data retrieved. See "UIEventDataTypeCode" on page 8-14.
dataString	xsd:string	The data sent from the underlying network node.
Returns		
Void.		
Exceptions		
-		

CHAPTER 9

Subscriber Profile

The following sections provide detailed information about the Subscriber Profile service capability:

Web Service

The subscriber profiles in are used for defining data on the subscriber, such as name, address, information on the terminal type and so on.

An applicationr with rights to use the subscriber profile service can perform the following tasks:

- Retrieve administrative and technical information for a subscriber
- Set administrative and technical information on a subscriber

getSubscriberProperty

Get a set of data, properties, for a subscriber in the subscriber profile database. Asynchronous request.

Table 9-1 getSubscriberProperty(endpoint, address, propertyTypes, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 9-10. An example is `http://www.acompany.com/axis/services/SubscriberProfileListener`
address	xsd:string	The address, in URI format (tel:<address>), of the subscriber to get properties. for.
propertyTypes	impl:ArrayOfPropertyTypes	See "ArrayOfPropertyTypes" on page 9-7. List of properties to fetch.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 9-1 getSubscriberProperty(endpoint, address, propertyTypes, serviceCode, requesterID)

getSubscriberPropertyReturn	xsd:int	ID of the request. This ID is supplied in getSubscriberPropertyResult to keep track of the different request-response pairs.
Exceptions		
SubscriberProfileException		See "SubscriberProfileException" on page 9-6.
GeneralException		See "GeneralException" on page 9-6.

getSubscriberPropertyWait

Get a set of data, properties, for a subscriber in the subscriber profile database. Synchronous request.

Table 9-2 getSubscriberPropertyWait(address, propertyTypes, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
address	xsd:string	The address, in URI format (tel:<address>), of the subscriber to get properties for.
propertyTypes	impl:ArrayOfPropertyTypes	See "ArrayOfPropertyTypes" on page 9-7. List of properties to fetch.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.

Table 9-2 getSubscriberPropertyWait(address, propertyTypes, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Returns		
getSubscriberPropertyWaitReturn	impl:ArrayOfProperty	See "ArrayOfProperty" on page 9-6. Array of the requested properties.
Exceptions		
SubscriberProfileException		See "SubscriberProfileException" on page 9-6.
GeneralException		See "GeneralException" on page 9-6.

setSubscriberProperty

Set a set of data, properties, for a subscriber in the subscriber profile database. Asynchronous request.

Table 9-3 setSubscriberProperty(endpoint, address, properties, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 9-10. An example is `http://www.acompany.com/axis/services/SubscriberProfileListener`
address	xsd:string	The address, in URI format (tel:<address>), of the subscriber to set properties. for.
properties	impl:ArrayOfProperty	See "ArrayOfProperty" on page 9-6. Properties to set.

Table 9-3 setSubscriberProperty(endpoint, address, properties, serviceCode, requesterID)

Parameter Name	Type	Description
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
setSubscriberPropertyReturn	xsd:int	ID of the request. This ID is supplied in setSubscriberProperty to keep track of the different request-response pairs.
Exceptions		
SubscriberProfileException		See "SubscriberProfileException" on page 9-6.
GeneralException		See "GeneralException" on page 9-6.

setSubscriberPropertyWait

Get a certain data, property, for a subscriber in the subscriber profile database. Synchronous request.

Table 9-4 setSubscriberPropertyWait(address, properties, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
address	xsd:string	The address, in URI format (tel:<address>), of the subscriber to set properties. for.
properties	impl:ArrayOfProperty	See "ArrayOfProperty" on page 9-6. List of properties to fetch.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.

Table 9-4 setSubscriberPropertyWait(address, properties, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
Void.		
Exceptions		
SubscriberProfileException		See "SubscriberProfileException" on page 9-6.
GeneralException		See "GeneralException" on page 9-6.

Exceptions

SubscriberProfileException

Exceptions of this type are raised when there are error conditions related to the Subcriber Profile Web Service. Other error conditions are reported using the exception GeneralException. See "SubscriberProfileException" on page 9-10 for a definition of the datatype.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

ArrayOfProperty

Array of Properties. See "ArrayOfProperty" on page 9-6 for information on each element in the array.

ArrayOfPropertyTypes

Array of PropertyTypes. See "ArrayOfPropertyTypes" on page 9-7 for information on each element in the array.

PaymentMethod

Defines a payment method and the related value.

Table 9-5

Name	Type	Description
paymentType	intf:PaymentType	See "PaymentType" on page 9-7.
value	xsd:anyType	See "PaymentType" on page 9-7 for which type to use. Depends on type defined in paymentType.

PaymentType

Type of payment method. Enumeration (xsd:string) with one of the following values.

Table 9-6

Name	Description
PAYMENT_TYPE_UNDEFINED	Undefined. Null.
CREDIT_CARD	Credit card. Use xsd:string in xsd:AnyType in PaymentMethod. Credit card number.
INVOICE	Invoice. Use xsd:short in xsd:AnyType in PaymentMethod. Customer number.

PropertyTypes

Type of property to set or get. Enumeration (xsd:string) with one of the following values.

Table 9-7

Value	Description
UNDEFINED	Undefined. Null.
NAME	Name.
ALIAS	Alias to be used to ensure anonymity between users of the application.
ADDRESS	Complete postal address.
HOME_PHONE	Home phone number.
OFFICE_PHONE	Office phone number.
PRIVATE_MAIL	Private e-mail address.
OFFICE_MAIL	Office e-mail address.
TERMINAL_ID	IMSI of the terminal.
TERMINAL_VENDOR	Vendor name, for example Ericsson, Nokia, or Siemens.
TERMINAL_MODEL	Model name of the terminal. For example T68i, 3210 or A55.
TERMINAL_SCREEN_SIZE	Screen size of the terminal. Character rows x columns.
COLOR_ENABLED_TERMINAL	Colour enabled-terminal. Yes/No.
MMS_ENABLED_TERMINAL	MMS enabled terminal. Yes/No.
FAX_NUMBER	Fax number.
GROUP_IDENTITY	For example family, office location or work group.
GENDER	Male/Female
BIRTH_DATE	Birth date. In format YYYY-MM-DD.
NATIONALITY	Nationality. Free text.
MOTHER_TOUNGE	Mother tongue. Free text.
CURRENCY	Currency. Free text.

Table 9-7

Value	Description
MISCELLANEOUS	Any type of additional information.
LAST_UPDATED	Date and time the account was last updated (read-only).
LAST_UPDATED_BY	The user that updated the account (read-only).
SUBSCRIPTION_TYPE	Type of subscription, Prepaid, Postpaid, Time Limited, or Free.
PAYMENT_METHOD	Payment method: Credit card or Invoice.
BALANCE	Account balance. (read-only).
APPLICATION_SUBSCRIPTIONS	List of subscribed applications.

RowCol

Table 9-8

Name	Type	Description
co	xsd:short	Column size.
row	xsd:short	Row size.

SubscriberGender

Defines the gender. Enumeration (xsd:string) with the following values.

Table 9-9

Value	Description
MALE	-
FEMALE	-

SubscriberProfileException

Holder for exceptions.

Table 9-10

Name	Type	Description
exceptionMessage	xsd:string	Exception Message in plaint text.
errorCode	xsd:int	Unique error code.

SubscriptionType

Defines different subscription types for a subscriber. Enumeration (xsd:string) with the following values.

Table 9-11

Value	Name
SUBSCRIPTION_TYPE_UNDEFINED	Undefined.
PRE_PAID	Pre-paid account.
POST_PAID	Post paid account.
FREE	Free.
TIME_LIMITED	Time limited

Listener interface

The Subscriber profile listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the location Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that a session is no longer valid. The application can not use the session no more.

Table 9-12 deactivate(id)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the session to be deactivated. The ID was returned when getSubscriberProperty or setSubscriberProperty was invoked.
Returns		
Void.		
Exceptions		
-		

getSubscriberPropertyError

Errors related to requests invoked by getSubscriberProperty are reported using this method.

Table 9-13

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getSubscriberProperty was invoked.
address	xsd:string	Address, in URI format (tel:<address>), of the subscriber whose properties were requested.
errorMsg	xsd:string	Error message.
Returns		
Void.		

Table 9-13

Parameter Name	Type	Description
Exceptions		
-		

getSubscriberPropertyResult

The result of a successful invocation of getSubscriberProperty.

Table 9-14 getSubscriberPropertyResult(id, address, properties)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getSubscriberProperty was invoked.
address	xsd:string	Address, in URI format (tel:<address>), of the subscriber whose properties were requested.
properties	impl:ArrayOfProperty	See "ArrayOfProperty" on page 9-6. Array containing the requested information.
Returns		
Void.		
Exceptions		
-		

setSubscriberPropertyError

Errors related to requests invoked by setSubscriberProperty are reported using this method.

Table 9-15 getSubscriberPropertyError(id, address, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when setSubscriberProperty was invoked.
address	xsd:string	Address, in URI format (tel:<address>), of the subscriber whose properties were requested.
errorMsg	xsd:string	Error message.
Returns		
Void.		
Exceptions		
-		

setSubscriberPropertyResult

The result of a successful invocation of setSubscriberProperty.

Table 9-16 setSubscriberPropertyResult(id, address)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getSubscriberProperty was invoked.
address	xsd:string	Address, in URI format (tel:<address>), of the subscriber whose properties were requested.
Returns		

Table 9-16 setSubscriberPropertyResult(id, address)

Parameter Name	Type	Description
Void.		
Exceptions		
-		

Complex data types

The data types are the same as the ones for the Subscriber profile Web Service, described in section "Complex data types" on page 9-6.

CHAPTER 10

User Location

The following sections provide detailed information about the User Location service capability:

Web Service

The user location Web Service provides an application with functions for retrieving the location of a mobile terminal.

getExtendedLocation

Retrieves a mobile terminal's location with extended data. Asynchronous request.

Table 10-1 getExtendedLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), to request location for.
altitudeRequested	xsd:boolean	Defines if altitude will be supplied in the location data.
priority	tns7:Priority	See "Priority" on page 10-19.
Priority of the request.		
requestedAccuracy	xsd:float	Unit is meters.
requestedLocationMethod	xsd:string	Depends on operator settings.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
type	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.

Table 10-1 getExtendedLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, serviceCode, requesterID)

Parameter Name	Type	Description
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getExtendedLocationReturn	xsd:int	ID of the request. This ID is supplied in "setExtendedLocation" on page 10-30 to keep track of the different request-response pairs.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

getExtendedLocationWait

Retrieves a mobile terminal's location with extended data and returns the location directly. Synchronous request.

Table 10-2 getExtendedLocationWait(addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the mobile terminals to be positioned.

Table 10-2 getExtendedLocationWait(addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
altitudeRequested	xsd:boolean	Defines if altitude will be supplied in the location data.
priority	tns7:Priority	See "Priority" on page 10-19. Priority of the request.
requestedAccuracy	xsd:float	Unit is metres.
requestedLocationMethod	xsd:string	Depends on operator settings.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
type	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-2 getExtendedLocationWait(addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
getExtendedLocationWaitReturn	impl:ArrayOfExtendedLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20.Array of locations, one for each terminal which location was requested.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

getGeoLocation

Retrieves a mobile terminal's location with geographical information, such as street address, included. Asynchronous request.

Table 10-3 getGeoLocation(endpoint, addresses, priority, requestedResponseTime, type, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), to request location for.

Table 10-3 getGeoLocation(endpoint, addresses, priority, requestedResponseTime, type, serviceCode, requesterID)

Parameter Name	Type	Description
priority	tns7:Priority	See "Priority" on page 10-19. Priority of the request.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
type	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getGeoLocationReturn	xsd:int	ID of the request. This ID is supplied in setGeoLocation to keep track of the different request-response pairs.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

getGeoLocationWait

Retrieves a mobile terminal's location with geographical information, such as street address, included and returns the location directly. Synchronous request.

Table 10-4 getGeoLocationWait(addresses, priority, requestedResponseTime, type, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the terminals to be positioned.
priority	tns7:Priority	See "Priority" on page 10-19.Priority of the request.
requestedAccuracy	xsd:float	Unit is meters.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
type	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-4 getGeoLocationWait(addresses, priority, requestedResponseTime, type, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
getGeoLocationWaitReturn	impl:ArrayOfGeoLocationResult	See "ArrayOfGeoLocationResult" on page 10-29. Array of locations, one for each terminal which location was requested.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

getLocation

Retrieves a mobile terminal's location. Asynchronous request.

Table 10-5 getLocation(endpoint, addresses, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the mobile terminals to be positioned.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-5 getLocation(endpoint, addresses, serviceCode, requesterID)

Parameter Name	Type	Description
getLocationReturn	xsd:int	ID of the request. This ID is supplied in setGeoLocation to keep track of the different request-response pairs.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

getLocationWait

Retrieves a mobile terminal's location and return the location directly. Synchronous request.

Table 10-6 getLocationWait(addresses, requestedResponseTime, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the mobile terminals to be positioned.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-6 getLocationWait(addresses, requestedResponseTime, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
getLocationWaitReturn	impl:ArrayOfLocationResult	See "ArrayOfLocationResult" on page 10-23. Array of locations, one for each terminal which location was requested.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

startPeriodicGeoLocation

Starts to retrieve mobile user's locations with geographical information, such as street address, included periodically. Asynchronous request.

Table 10-7 startPeriodicGeoLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedLocationResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`

Table 10-7 startPeriodicGeoLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedLocationResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the mobile terminals to be positioned.
altitudeRequested	xsd:boolean	Defines if altitude will be supplied in the location data.
Priority	tns7:Priority	See "Priority" on page 10-19. Priority of the request.
requestedAccuracy	xsd:float	Requested accuracy given in metres.
requestedLocationMethod	xsd:string	Depends on operator settings.
LocationResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
wespaLocationtype	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.
reportingInterval	xsd:long	The reporting interval in milliseconds.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-7 startPeriodicGeoLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedLocationResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
startPeriodicGeoLocationReturn	xsd:int	ID of the session. This ID is supplied in "setPeriodicGeoLocation" on page 10-34 to keep track of the request and the associated responses. The ID is also used when stopping the session.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

startPeriodicLocation

Starts to retrieve mobile user's locations with extended location data included periodically. Asynchronous request.

Table 10-8 startPeriodicLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`

Table 10-8 startPeriodicLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>), of the mobile terminals to be positioned.
altitudeRequested	xsd:boolean	Defines if altitude will be supplied in the location data.
priority	tns7:Priority	See "Priority" on page 10-19. Priority of the request.
requestedAccuracy	xsd:float	Requested accuracy given in meters.
requestedLocationMethod	xsd:string	Depends on operator settings.
requestedResponseTime	impl:LocationResponseTime	See "LocationResponseTime" on page 10-18.
type	impl:LocationType	See "LocationType" on page 10-17. The type of location requested.
reportingInterval	xsd:long	The reporting interval in milliseconds.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
startPeriodicLocation	xsd:int	ID of the request. This ID is supplied in setPeriodicLocation to keep track of the request and the associated responses.
Exceptions		

Table 10-8 startPeriodicLocation(endpoint, addresses, altitudeRequested, priority, requestedAccuracy, requestedLocationMethod, requestedResponseTime, type, reportingInterval, serviceCode, requesterID)

Parameter Name	Type	Description
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

startTriggeredLocationReporting

Starts a triggered location tracking reporting session. When a mobile terminal enters a certain area, a report is sent to the application. Asynchronous request.

Table 10-9 startTriggeredLocationReporting(endpoint, addresses, trigger, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 10-29. An example is `http://www.acompany.com/axis/services/LocationListener`
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 10-29. The addresses, in URI format (tel:<address>) of the mobile terminals to be positioned.
trigger	impl:LocationTrigger	See "LocationTrigger" on page 10-19. Definition of the area or geographical location for which a location report shall be sent to the application.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		

Table 10-9 startTriggeredLocationReporting(endpoint, addresses, trigger, serviceCode, requesterID)

Parameter Name	Type	Description
startPeriodicLocation	xsd:int	ID of the request. This ID is supplied in setPeriodicLocation to keep track of the request and the associated responses.
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

stopPeriodicGeoLocation

Stops a periodic geographical location tracking session. Synchronous request.

Table 10-10 stopPeriodicGeoLocation(id)

Parameter Name	Type	Description
Input		
id	xsd:int	Identifier for the session. See "startPeriodicGeoLocation" on page 10-10 for information on how to start the session.
Returns		
Void.		
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

stopPeriodicLocation

Stops a periodic location tracking session. Synchronous request.

Table 10-11 stopPeriodicLocation(id)

Parameter Name	Type	Description
Input		
id	xsd:int	Identifier for the session. See "startPeriodicLocation" on page 10-12 for information on how to start the session.
Returns		
Void.		
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

stopTriggeredLocationReporting

Stops a location triggered location tracking session. Synchronous request.

Table 10-12 stopTriggeredLocationReporting(id)

Parameter Name	Type	Description
Input		
id	xsd:int	Identifier for the session. See "startTriggeredLocationReporting" on page 10-14 for information on how to start the session.
Returns		
Void.		
Exceptions		
LocationException		See "LocationException" on page 10-17.
GeneralException		See "GeneralException" on page 10-17.

Exceptions

LocationException

Exceptions of this type are raised when there are error conditions related to the location Web Service. Other error conditions are reported using the exception GeneralException.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

LocationType

Type of location requested. Enumeration (xsd:string) with one of the following values.

Table 10-13

Value	Description
CURRENT	Current location of the mobile terminal.
CURRENT_OR_LAST_KNOWN	Current location of the mobile terminal, or last known if current is not available.
INITIAL	Initial location of the mobile terminal. For future use.

LocationResponseTime

Table 10-14

Name	Value	Description
responseTimeIndicator	intf:LocationResponseTimeIndicator	See "LocationResponseTimeIndicator" on page 10-18.
timerValue	xsd:int	Application-defined response time. Valid only in combination when responseTimeIndicator is set to use user-defined value. See "LocationResponseTimeIndicator" on page 10-18. Depends on the network.

LocationResponseTimeIndicator

Indicates the accepted response times (QoS). Enumeration (xsd:string) with one of the following values.

Table 10-15

Value	Description
DELAY_TOLERANT	Used for priority handling and queuing in the network.
LOW_DELAY	Used for priority handling and queuing in the network.
NO_DELAY	Used for priority handling and queuing in the network.
USE_TIMER_VALUE	Sets the timer to the time indicated in LocationResponseTime used.

Priority

Indicates the desired priority of the request (QoS). Enumeration (xsd:string) with one of the following values.

Table 10-16

Value	Description
LOW	-
HIGH	-
NORMAL	-

LocationTrigger

Defines the geographical area and the event type that shall trigger a location report. Event type is either when the terminal enters or exits the area.

Table 10-17

Name	Type	Description
location	intf:LocationTriggerLocation	See "LocationTriggerLocation" on page 10-19.
criteria	intf:LocationTriggerCriteria	See "LocationTriggerCriteria" on page 10-20.

LocationTriggerLocation

Defines the type of area and the data related to that area.

Table 10-18

Name	Type	Description
triggerLocationTypes	intf:TriggerLocationTypes	See "TriggerLocationTypes" on page 10-27.
value	xsd:anyType	Either LocationTriggerLongLat or GeoLocation depending on type defined in triggerLocationTypes.

LocationTriggerCriteria

Defines when the location report shall be distributed to the application. Enumeration (xsd:string) with the following values.

Table 10-19

Value	Description
ENTERING_AREA	Used to trigger the response when the mobile terminal enters the area defined in LocationTrigger.
LEAVING_AREA	Used to trigger the response when the mobile terminal leaves the area defined in LocationTrigger.

ArrayOfExtendedLocationResult

Array of locations. See "ExtendedLocationResult" on page 10-21 for information on the individual elements in the array.

ExtendedLocationResult

Table 10-20

Name	Type	Description
address	xsd:string	The address, in URI format (tel:<address>), of the located mobile terminal.
status	intf:LocationStatusCode	See "LocationStatusCodes" on page 10-21.
theLocation	intf:ExtendedLocation	See "ExtendedLocation" on page 10-22.

LocationStatusCodes

Information about the outcome of a location request. Enumeration (xsd:string) with one of the following values.

Table 10-21

Value	Description
OK	The location request was successful.
SYSTEM_FAILURE	The location request failed due to failure in the network.
UNAUTHORIZED_NETWORK	The location request failed due to that the network did not accept a location request to be performed.
UNAUTHORIZED_APPLICATION	The location request failed due to that the application was unauthorized to perform the request.
UNKNOWN_SUBSCRIBER	The location request failed due to that the address of the mobile terminal was unknown.
ABSENT_SUBSCRIBER	The location request failed due to that the position of the mobile terminal is unknown.
POSITION_METHOD_FAILURE	The location request failed due to that the requested positioning method is inaccessible or not supported by the network.

ExtendedLocation

Location data, extended with information in altitude and terminal type.

Table 10-22

Name	Type	Description
theLocation	intf:Location	See "Location" on page 10-23.
altitudePresent	xsd:boolean	True if altitude is present in the data.
altitude	xsd:float	Altitude given in metres.
uncertaintyAltitude	xsd:float	Uncertainty indicator of the altitude given in meeteres.
theTerminalType	tns7:TerminalType	See "TerminalType" on page 10-22.
timeStamp	xsd:string	Time when the terminal was positioned. Format is YYYY-MM-DD HH:MM:SS.mmm.
locationMethod	xsd:string	Method used when the mobile terminal was positioned if supported by the network.

TerminalType

Information on the terminal with the given address. Enumeration (xsd:string) with one of the following values.

Table 10-23

Value	Description
TT_FIXED	Fixed terminal.
TT_IP	IP terminal.
TT_MOBILE	Mobile terminal.
TT_UNKNOWN	Unknown terminal type.

ArrayOfLocationResult

Array of locations. See "LocationResult" on page 10-23 for information on the individual elements in the array.

LocationResult

Table 10-24

Name	Type	Description
address	xsd:string	The MSISDN, in URI format (tel:<address>), of the located mobile terminal.
status	intf:LocationStatusCode	See "LocationStatusCodes" on page 10-21.
theLocation	intf:Location	See "Location" on page 10-23.

Location

Location data with longitude, latitude, and indicator of the uncertainty shape.

Table 10-25

Name	Type	Description
latitude	xsd:float	-
longitude	xsd:float	-
shape	intf:LocationUncertaintyShape	See "LocationUncertaintyShape" on page 10-24.

LocationUncertaintyShape

Defines a location uncertainty shape.

Table 10-26

Name	Type	Value
locationUncertaintyShapeType	intf:LocationUncertaintyShapeTypes	See "LocationUncertaintyShapeTypes" on page 10-24.
value	xsd:anyType	Depends on what is specified in locationUncertaintyShapeType. See "LocationUncertaintyShapeTypes" on page 10-24 for mapping between data type and locationUncertaintyShapeType.

LocationUncertaintyShapeTypes

For a description of the uncertainty shapes see "Location Uncertainty shapes" on page 10-39.

Enumeration (xsd:string) within of the following values.

Table 10-27

Value	Description
CIRCLE	See "Circle uncertainty shapes" on page 10-39.
CIRCLE_SECTOR	See "Circle uncertainty shapes" on page 10-39.
CIRCLE_ARC_STRIPE	See "Circle uncertainty shapes" on page 10-39.
ELLIPSE	See "Ellipse uncertainty shapes" on page 10-40.
ELLIPSE_SECTOR	See "Ellipse uncertainty shapes" on page 10-40.
ELLIPSE_ARC_STRIPE	See "Ellipse uncertainty shapes" on page 10-40.
UNDEFINED	Allows to use thee default location method of the network.

LocationUncertaintyShapeCircle

Defines the circle uncertainty shape. See Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

Table 10-28

Name	Type	Value
radius	xsd:float	Radius of the circle given in meters.

LocationUncertaintyShapeCircleArcStripe

Defines the circle arc stripe uncertainty shape. See Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

Table 10-29

Name	Type	Description
circleSector	intf:LocationUncertaintyShapeCircleSector	The circle sector. See “LocationUncertaintyShapeCircleSector” on page 10-25.
innerRadius	xsd:float	Radius of the inner circle given in metres.

LocationUncertaintyShapeCircleSector

Defines the circle sector uncertainty shape. See Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

Table 10-30

Name	Type	Description
circle	intf:LocationUncertaintyShapeCircle	Radius of the circle. See “LocationUncertaintyShapeCircle” on page 10-25.

Table 10-30

Name	Type	Description
segmentEndAngle	xsd:float	End angle of the circle sector given in degrees.
segmentStartAngle	xsd:float	Start angle of the circle sector given in degrees.

LocationUncertaintyShapeEllipse

Defines the ellipse uncertainty shape. See Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

Table 10-31

Name	Type	Description
angle	xsd:float	Angle of the semi major axis given in degrees.
semiMajor	xsd:float	Size of the semi major axis given in meters.
semiMinor	xsd:float	Size of the semi minor axis given in meters.

LocationUncertaintyShapeEllipseArcStripe

Defines the ellipse arc stripe uncertainty shape. See Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

Table 10-32

Name	Type	Description
ellipseSector	intf:LocationUncertaintyShapeEllipseSector	Definition of the ellipse. See “LocationUncertaintyShapeEllipseSector” on page 10-27.

Table 10-32

Name	Type	Description
innerSemiMajor	xsd:float	Size of semi major axis of the inner ellipse defining the stripe. Given in meters.
innerSemiMinor	xsd:float	Size of semi minor axis of the inner ellipse defining the stripe. Given in meters.

LocationUncertaintyShapeEllipseSector

Defines the ellipse sector uncertainty shape. See Figure 10-2, “Ellipse uncertainty shape definitions,” on page 10-41.

Table 10-33

Name	Type	Description
ellipse	intf:LocationUncertaintyShapeEllipse	Defines the ellipse. See “LocationUncertaintyShapeEllipse” on page 10-26.
segmentEndAngle	xsd:float	Defines the end angle of the sector, given in degrees.
segmentStartAngle	xsd:float	Defines the start angle of the sector given in degrees.

TriggerLocationTypes

Definition of which geographical area definition method to use for triggered user location.

Enumeration (xsd:string) with the following values.

Table 10-34

Value	Description
LONG_LAT	Area definition method is based on longitude and latitude. See "LocationTriggerLongLat" on page 10-28.
GEO_DATA	Area definition method is based on abstract geographical definitions such as street address, ZIP-code and so on. See "GeoLocation" on page 10-28.

LocationTriggerLongLat

Definition of a geographical area. See "Ellipse uncertainty shapes" on page 10-40. The area is defined by an ellipse.

Table 10-35

Name	Type	Description
Longitude	xsd:float	Longitude of the base point.
Latitude	xsd:float	Latitude of the base point.
AreaSemiMajor	xsd:float	Size of semi major axis ellipse given in meters.
AreaSemiMinor	xsd:float	Size of semi minor axis ellipse given in meters.
AngleOfSemiMajor	xsd:float	Angel of semi major axis given in degrees.

GeoLocation

Definition of a geographical area. Defined by names representing a geographical location.

Table 10-36

Value	Type	Decription
streetAddress	xsd:string	Street address.
county	xsd:string	County.
zipCode	xsd:string	ZIP code.

Table 10-36

Value	Type	Decription
city	xsd:string	City.
state	xsd:string	State.
country	xsd:string	Country.
area	xsd:string	Operator-defined.
network	xsd:string	Operator-defined.

ArrayOfGeoLocationResult

Array of geographical locations. See "GeoLocationResult" on page 10-29 for information on the individual records in the array.

GeoLocationResult

Table 10-37

Name	Type	Description
address	xsd:string	The address, in URI format (tel:<address>), of the located terminal.
status	intf:LocationStatusCode	See "LocationStatusCodes" on page 10-21.
theLocation	intf:GeoLocation	See "GeoLocation" on page 10-28.

ArrayOf_xsd_string

Array of strings.

Listener interface

The location listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the location Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that a session is no longer valid. The application can not use the session no more.

Table 10-38 deactivate(id)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the session to be deactivated. The ID was returned when getExtendedLocation, getGeoLocation, or getLocation was invoked.
Returns		
Void.		
Exceptions		
-		

setExtendedLocation

Responses to requests invoked by "getExtendedLocation" on page 10-2 are reported using this method.

Table 10-39 setExtendedLocation(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getExtendedLocation was invoked.
result	impl:ArrayOfExtendedLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20. The requested location data.
Returns		
Void.		

Table 10-39 setExtendedLocation(id, result)

Parameter Name	Type	Description
Exceptions		
-		

setExtendedLocationError

Errors related to requests invoked by getExtendedLocation are reported using this method.

Table 10-40 setExtendedLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getExtendedLocation was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		
Void.		
Exceptions		
-		

setGeoLocation

Responses to requests invoked by getGeoLocation are reported using this method.

Table 10-41 setGeoLocation(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getGeoLocation was invoked.
result	impl:ArrayOfGeoLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20. The requested location data.
Returns		
Void.		
Exceptions		
-		

setGeoLocationError

Errors related to requests invoked by getGeoLocation are reported using this method.

Table 10-42 setGeoLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getGeoLocation was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		

Table 10-42 setGeoLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Void.		
Exceptions		
-		

setLocation

Responses to requests invoked by getLocation are reported using this method.

Table 10-43 setLocation(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getLocation was invoked.
result	impl:ArrayOfLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20. The requested location data.
Returns		
Void.		
Exceptions		
-		

setLocationError

Errors related to requests invoked by getLocation are reported using this method.

Table 10-44 setLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getLocation was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		
Void.		
Exceptions		
-		

setPeriodicGeoLocation

Responses to requests invoked by startPeriodicGeoLocation are reported using this method.

Table 10-45 setLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when "startPeriodicGeoLocation" on page 10-10 was invoked.
result	impl:ArrayOfGeoLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20. The requested location data.
Returns		

Table 10-45 setLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Void.		
Exceptions		
-		

setPeriodicGeoLocationError

Errors related to requests invoked by startPeriodicGeoLocation are reported using this method.

Table 10-46 setPeriodicGeoLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when startPeriodicGeoLocation was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		
Void.		
Exceptions		
-		

setPeriodicLocation

Responses to requests invoked by startPeriodicLocation are reported using this method.

Table 10-47 setPeriodicLocation(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when startPeriodicLocation was invoked.
result	impl:ArrayOfLocationResult	See "ArrayOfExtendedLocationResult" on page 10-20. The requested location data.
Returns		
Void.		
Exceptions		
-		

setPeriodicLocationError

Errors related to requests invoked by startPeriodicLocation are reported using this method.

Table 10-48 setPeriodicLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when startPeriodicLocation was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		

Table 10-48 setPeriodicLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Void.		
Exceptions		
-		

setTriggeredLocation

Responses to requests invoked by startTriggeredLocationReporting are reported using this method.

Table 10-49 setTriggeredLocation(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when startTriggeredLocationReporting was invoked.
result	impl:ArrayOfTriggeredLocationResult	See "ArrayOfTriggeredLocationResult" on page 10-39. The requested location data.
Returns		
Void.		
Exceptions		
-		

setTriggeredLocationError

Errors related to requests invoked by startTriggeredLocationReporting are reported using this method.

Table 10-50 setTriggeredLocationError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when startTriggeredLocationReporting was invoked.
errorCode	xsd:int	Code representing the error.
errorMsg	xsd:string	Error description.
Returns		
Void.		
Exceptions		
-		

Exceptions

-

Complex data types

Below are the complex data types defined. Only the ones specific for the listener interface are defined below, the WSDL file contains additional data definitions, but they are the same as for location Web Service and are described in section "Complex data types" on page 10-17.

ArrayOfTriggeredLocationResult

Array of locations. See "TriggeredLocationResult" on page 10-39 for information on the individual elements in the array.

TriggeredLocationResult

Table 10-51

Name	Type	Description
address	xsd:string	The MSISDN, in URI format (tel:<address>), of the located mobile terminal.
status	intf:LocationStatusCode	See "LocationStatusCodes" on page 10-21.
theLocation	intf:Location	See "Location" on page 10-23.
criteria	intf:TriggerCriteria	See "TriggerCriteria" on page 10-39.

TriggerCriteria

Defines when a a triggered location report shall be distributed to the application. Enumeration (xsd:string) with one of the following values.

Table 10-52

Value	Description
UL_ENTERING_AREA	Used when the mobile terminal enters a defined area.
UL_LEAVING_AREA	Used when ten mobile terminal leaves a defined area.

Location Uncertainty shapes

Circle uncertainty shapes

The circle uncertainty shapes are:

- Circle
- Circle sector

- Circle arc

The circle sector is an extended case of the circle, and the circle arc is a extended case of the circle sector, see Figure 10-1, “Circle uncertainty shape definitions,” on page 10-40.

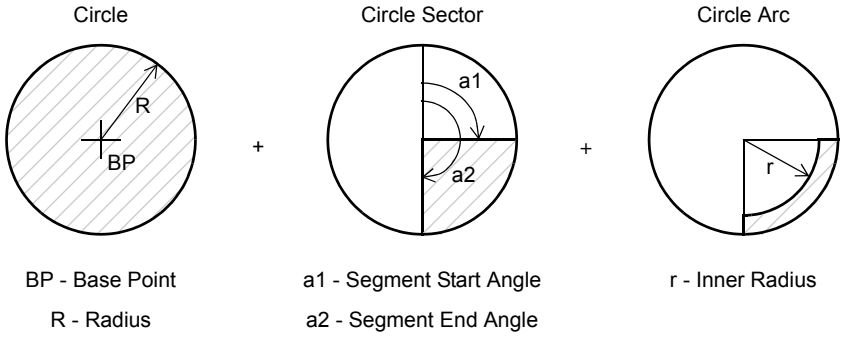

Figure 10-1 Circle uncertainty shape definitions

Ellipse uncertainty shapes

The ellipse uncertainty shapes are:

- Ellipse
- Ellipse sector
- Ellipse arc

The ellipse sector is an extended case of the ellipse, and the ellipse arc is a extended case of the ellipse sector, see Figure 10-2, “Ellipse uncertainty shape definitions,” on page 10-41.

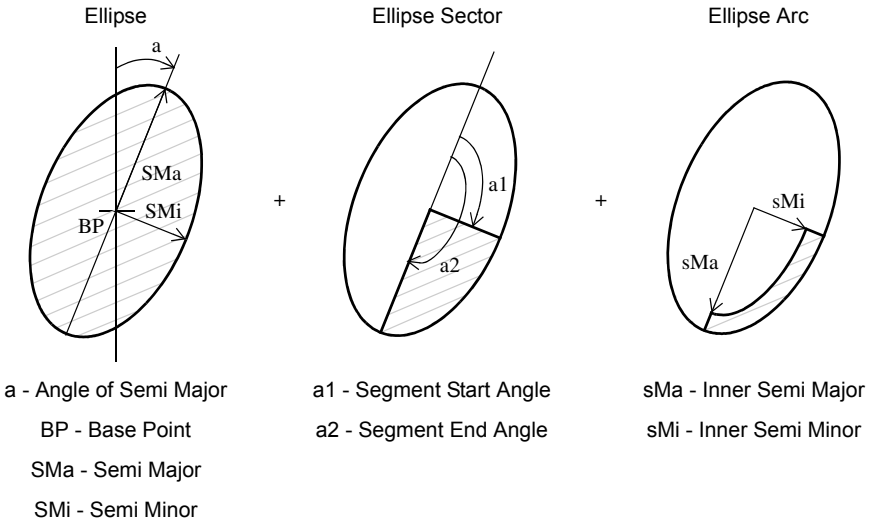

Figure 10-2 Ellipse uncertainty shape definitions

Terminal altitude

If the terminal's altitude is provided, the actual terminal altitude is somewhere within a span defined by the provided altitude value and two times the altitude uncertainty, see Figure 10-3, "Terminal altitude definition," on page 10-42.

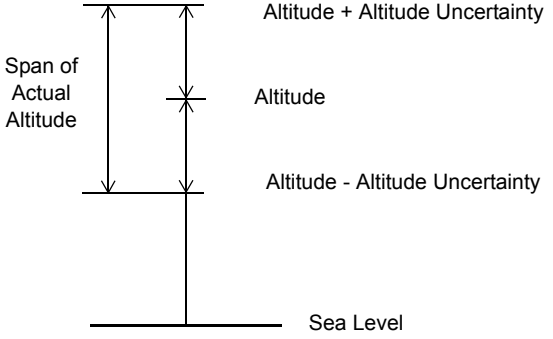

Figure 10-3 Terminal altitude definition

A positive altitude value means above sea level, whereas a negative value means below sea level.

CHAPTER 11

User Status

The following sections provide detailed information about the User Status service capability:

Web Service

The User Status Web Service provides an application with functions for retrieving the status of a telephony terminal.

getStatus

Get the status of one or more telephony terminals. Asynchronous request.

Table 11-1 getStatus(endpoint, addresses, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 11-9. An example is `http://www.acompany.com/axis/services/UserStatusListener`
addresses	mpl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 11-6. List of addresses, in URI format (tel:<address>), of the telephony terminals to retrieve the status for.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getStatusReturn	xsd:int	ID of the request. This ID is supplied in setStatus to keep track of the different request-response pairs.
Exceptions		
UserStatusException		See "UserStatusException" on page 11-5.
GeneralException		See "GeneralException" on page 11-6.

getStatusWait

Get the status of one or more telephony terminals. Synchronous request.

Table 11-2 getStatusWait(addresses, waitTimeoutSeconds, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 11-6. List of addresses, in URI format (tel:<address>), of the telephony terminals to retrieve the status for.
waitTimeoutSeconds	xsd:int	The time, in seconds, to wait for a response. If the response is not delivered within this time frame, and exception is thrown.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
getStatusWaitReturn	impl:ArrayOfStatusResult	See "ArrayOfStatusResult" on page 11-6. Array of the requested statuses.
Exceptions		
UserStatusException		See "UserStatusException" on page 11-5.
GeneralException		See "GeneralException" on page 11-6.

startTriggeredStatusReporting

Starts to retrieve the status of one or more telephony terminals periodically. Asynchronous request.

Table 11-3 startTriggeredStatusReporting(endpoint, addresses, serviceCode, requesterID)

Parameter Name	Type	Description
Input		
endpoint	xsd:string	The URL to the Web Service that implements the listener interface. See "Listener interface" on page 11-9. An example is `http://www.acompany.com/axis/services/UserStatusListener`
addresses	impl:ArrayOf_xsd_string	See "ArrayOf_xsd_string" on page 11-6. List of addresses, in URI format (tel:<address>), of the telephony terminals to retrieve the status for.
serviceCode	xsd:string	Used for charging purposes. Defined by the operator.
requesterID	xsd:string	The application ID as given by the operator.
Returns		
startTriggeredStatusReportingReturn	xsd:int	ID of the request. This ID is supplied in setStatus to keep track of the different request-response pairs and to stop the reporting, see stopTriggeredStatusReporting.
Exceptions		

Table 11-3 startTriggeredStatusReporting(endpoint, addresses, serviceCode, requesterID)

Parameter Name	Type	Description
UserStatusException		See "UserStatusException" on page 11-5.
GeneralException		See "GeneralException" on page 11-6.

stopTriggeredStatusReporting

Stops a periodic user status reporting session. Synchronous request.

Table 11-4 stopTriggeredStatusReporting(id)

Parameter Name	Type	Description
Parameters		
id	xsd:int	Identifier for the session. See "startTriggeredStatusReporting" on page 11-4 for information on how to start the session.
Returns		
Void.		
Exceptions		
UserStatusException		See "UserStatusException" on page 11-5.
GeneralException		See "GeneralException" on page 11-6.

Exceptions

UserStatusException

Exceptions of this type are raised when there are error conditions related to the User Status Web Service. Other error conditions are reported using the exception GeneralException.

GeneralException

This exception is raised when the applications session has expired or there are communication problems with the underlying platform.

Complex data types

ArrayOf_xsd_string

Table 11-5

Name	Description
Array of xsd:string	Each element holds the MSISDN of a terminal.

ArrayOfStatusResult

Table 11-6

Name	Description
Array of StatusResult	A set of results of a user status request. See "StatusResult" on page 11-6.

StatusResult

The result of a user status request for an individual telephony terminal.

Table 11-7

Name	Type	Value
address	xsd:string	The telephony terminals address in URI format (tel:<address>).
reqStatus	impl:StatusCode	See "StatusCode" on page 11-7.
userStatus	impl:Status	See "Status" on page 11-8.

StatusCode

Indicates the outcome of the request. Enumeration (xsd:string) with one of the following values.

Table 11-8

Value	Description
OK	The request was successful.
SYSTEM_FAILURE	The request failed due to system failure.
UNAUTHORIZED_NETWORK	The request failed due to that the terminal belongs to network that does not accept status requests.
UNAUTHORIZED_APPLICATION	The request failed due to that the application was not authorized to perform a status request.
UNKNOWN_SUBSCRIBER	The request failed due to that the telephony terminal whose status was requested is unknown.
ABSENT_SUBSCRIBER	The request failed due to that the telephony terminal is not within the home network.
STATUS_METHOD_FAILURE	The request failed due to that the status request was performed using an illegal method.

Status

Table 11-9

Name	Type	Description
aStatusIndicator	impl:StatusIndicator	See "StatusIndicator" on page 11-8.
aTerminalType	tns7:TerminalType	"TerminalType" on page 11-8.

StatusIndicator

Indicates the status of the telephony terminal. Enumeration (xsd:string) with one of the following values:

Table 11-10

Value	Description
REACHABLE	The terminal is reachable.
NOT_REACHABLE	The terminal is not reachable.
BUSY	The terminal is busy.

TerminalType

Indicates the type of terminal whose status was requested. Enumeration (xsd:string) with one of the following values.

Table 11-11

Value	Description
TT_FIXED	Connected to the fixed network.
TT_IP	Connected to the IP-network.
TT_MOBILE	Mobile terminal.
TT_UNKNOWN	Unknown terminal type.

Listener interface

The User Status listener interface defines the methods that the underlying platform invokes on a Web Service that is implemented by an application. When an application performs asynchronous requests from the User Status Web Service, the responses are delivered according to this interface.

deactivate

Used by the underlying system to inform the application that a session is no longer valid. The application can not use the session no more.

Table 11-12 deactivate(id)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the session to be deactivated. The ID was returned when getStatus was invoked.
Returns		
Void.		
Exceptions		
-		

setStatusError

Errors related to requests invoked by getStatus are reported using this method.

Table 11-13 setStatusError(id, errorCode, errorMsg)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getStatus was invoked.

Table 11-13 setStatusError(id, errorCode, errorMsg)

Parameter Name	Type	Description
errorCode	xsd:int	Error code.
errorMsg	xsd:string	Error message.
Returns		
Void.		
Exceptions		
-		

setStatus

The result of a successful invocation of getStatus.

Table 11-14 setStatus(id, result)

Parameter Name	Type	Description
Input		
id	xsd:int	ID of the request. The ID was returned when getStatus was invoked.
result	impl:ArrayOfStatusResult	See ArrayOfStatusResult. Status information of the terminals whose status were requested.
Returns		
Void.		
Exceptions		
-		

Complex data types

The data types are the same as the ones for the User Status Web Service, described in section “Complex data types” on page 11-6.