

**3GPP TSG CN Plenary Meeting #14
Kyoto, Japan, 12-14 December 2001**

NP-010598

Source: CN5 (OSA)
Title: Rel-4 CRs 29.198-05
Agenda item: 8.5
Document for: Decision

Doc-1st-Level	Spec	CR	Pha	Subject	Cat	Ver Cur	Ver -New	Doc-2nd-Level	Workitem
NP-010598	29.198-05	003	Rel-4	Replace Out Parameters with Return Types	F	4.2.0	4.3.0	N5-010565	OSA1
NP-010598	29.198-05	004	Rel-4	Correction of description of sendInfoRes()	F	4.2.0	4.3.0	N5-011138	OSA1
NP-010598	29.198-05	005	Rel-4	Correction to handling of deassign on related object	F	4.2.0	4.3.0	N5-011140	OSA1
NP-010598	29.198-05	006	Rel-4	Correction to Exceptions Raised in UI	F	4.2.0	4.3.0	N5-011243	OSA1
NP-010598	29.198-05	007	Rel-4	Correction to values of TpUllInfoType	F	4.2.0	4.3.0	N5-011271	OSA1

CR-Form-v4

CHANGE REQUEST

⌘ **29.198-05 CR 003** ⌘ ev **-** ⌘ Current version: **4.2.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: ⌘ (U)SIM ME/UE Radio Access Network Core Network

Title:	⌘ Replacing Out Parameters with Return Types		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 19/07/2001
Category:	⌘ F	Release:	⌘ REL-4
	<i>Use one of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		<i>Use one of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ At CN5 and CN it was agreed that Out-parameters should be removed from methods as a means of returning information, to be replaced by Return Types, in line with commonly used programming practice
Summary of change:	⌘ For each method, replace the return parameter TpResult with: 'void' if the method has no out-parameter; or the type of the out-parameter if the method has an out-parameter, and delete the out-parameter from the method.
Consequences if not approved:	⌘ If this particular CR is not agreed, TS 29.198-5 is out of sync. with the other parts of TS 29.198. If the related batch of CRs is not agreed, OSA will have a limited acceptance among the application development community, since it will be more difficult to implement. This presents a risk to the return on investment in development of OSA.

Clauses affected:	⌘ 7, 8, Annex B	
Other specs affected:	⌘ <input checked="" type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘ All other parts of TS 29.198 Rel-4
Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

7 The Service Interface Specifications

7.1 Interface Specification Format

This section defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described below.

7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name `Ip<name>`. The callback interfaces to the applications are denoted by classes with name `IpApp<name>`. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name `IpSvc<name>`, while the Framework interfaces are denoted by classes with name `IpFw<name>`

7.1.2 Method descriptions

Each method (API method “call”) is described. All methods in the API return a value of type `TpResult`, indicating, amongst other things, if the method invocation was successfully executed or not.

Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a 'Req' suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a 'Res' or 'Err' suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant `IpApp<name>` or `IpSvc<name>` interfaces to provide the callback mechanism.

7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as 'in' represent those that must have a value when the method is called. Those described as 'out' are those that contain the return result of the method when the method returns.

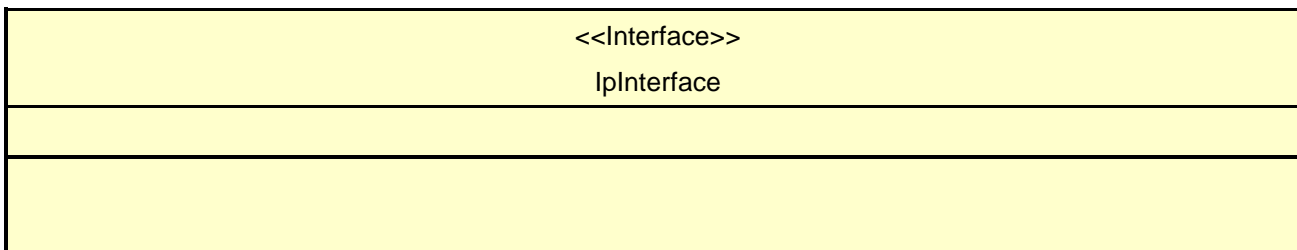
7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

7.2 Base Interface

7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.



7.3 Service Interfaces

7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

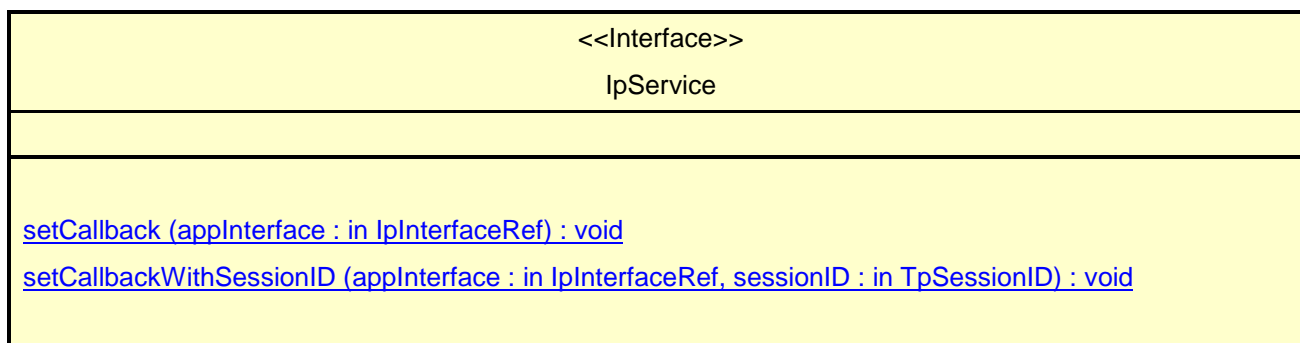
The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

7.4 Generic Service Interface

7.4.1 Interface Class IpService

Inherits from: IpInterface

All service interfaces inherit from the following interface.



Method

setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionID's.

Parameters

appInterface : in IpInterfaceRef

Specifies a reference to the application interface, which is used for callbacks

Raises

TpCommonExceptions

Method

setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not uses SessionID's.

Parameters

appInterface : in IpInterfaceRef

Specifies a reference to the application interface, which is used for callbacks

sessionID : in TpSessionID

Specifies the session for which the service can invoke the application's callback interface.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID

8 Generic User Interaction Interface Classes

The Generic User Interaction Service interface (GUI) is used by applications to interact with end users. The GUI is represented by the IpUIManager, IpUI and IpUICall interfaces that interface to services provided by the network. To handle responses and reports, the developer must implement IpAppUIManager and IpAppUI interfaces to provide the callback mechanism.

8.1 Interface Class IpUIManager

Inherits from: IpService.

This interface is the 'service manager' interface for the Generic User Interaction Service and provides the management functions to the Generic User Interaction Service.

<<Interface>> IpUIManager
<p> createUI (appUI : in IpAppUIRef, userAddress : in TpAddress) : TpUIIdentifier createUICall (appUI : in IpAppUICallRef, uiTargetObject : in TpUITargetObject) : TpUICallIdentifier createNotification (appUIManager : in IpAppUIManagerRef, eventCriteria : in TpUIEventCriteria) : TpAssignmentID destroyNotification (assignmentID : in TpAssignmentID) : void changeNotification (assignmentID : in TpAssignmentID, evenCriteria : in TpUIEventCriteria) : void getNotification () : TpUIEventCriteriaResultSet </p>

Method

createUI ()

[This method is used to create a new user interaction object for non-call related purposes](#)

[Results: userInteraction](#)

[Specifies the interface and sessionID of the user interaction created.](#)

Parameters

appUI : in IpAppUIRef

Specifies the application interface for callbacks from the user interaction created.

userAddress : in TpAddress

Indicates the end-user with whom to interact.

Returns

[TpUIIdentifier](#)

Raises

TpCommonExceptions, P_INVALID_NETWORK_STATE

Method

createUICall()

[This method is used to create a new user interaction object for call related purposes.](#)

[The user interaction can take place to the specified party or to all parties in a call. Note that for certain implementation user interaction can only be performed towards the controlling call party, which shall be the only party in the call.](#)

[Returns: userInteraction](#)

[Specifies the interface and sessionID of the user interaction created.](#)

Parameters

appUI : in IpAppUICallRef

Specifies the application interface for callbacks from the user interaction created.

uiTargetObject : in TpUITargetObject

Specifies the object on which to perform the user interaction. This can either be a Call, Multi-party Call or call leg object.

Returns

[TpUICallIdentifier](#)

Raises

TpCommonExceptions, P_INVALID_NETWORK_STATE

Method

createNotification()

[This method is used by the application to install specified notification criteria, for which the reporting is implicitly activated. If some application already requested notifications with criteria that overlap the specified criteria, the request is refused with P_INVALID_CRITERIA.](#)

[The criteria are said to overlap if both originating and terminating ranges overlap and the same number plan is used and the same servicecode is used.](#)

If the same application requests two notifications with exactly the same criteria but different callback references, the second callback will be treated as an additional callback. This means that the callback will only be used in case when the first callback specified by the application is unable to handle the reportNotification (e.g., due to overload or failure).

Returns: assignmentID

Specifies the ID assigned by the generic user interaction manager interface for this newly installed notification criteria.

Parameters

appUIManager : in IpAppUIManagerRef

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

eventCriteria : in TpUIEventCriteria

Specifies the event specific criteria used by the application to define the event required, like user address and service code.

Returns

TpAssignmentID

Raises

TpCommonExceptions,P_INVALID_CRITERIA

Method

destroyNotification()

This method is used by the application to destroy previously installed notification criteria via the createNotification method.

Parameters

assignmentID : in TpAssignmentID

Specifies the assignment ID given by the generic user interaction manager interface when the previous createNotification() was called. If the assignment ID does not correspond to one of the valid assignment IDs, the framework will return the error code P_INVALID_ASSIGNMENT_ID.

Raises

TpCommonExceptions,P_INVALID_ASSIGNMENT_ID

Method

changeNotification()

This method is used by the application to change the event criteria introduced with createNotification method. Any stored notification request associated with the specified assignmentID will be replaced with the specified events requested.

*Parameters***assignmentID : in TpAssignmentID**

Specifies the ID assigned by the manager interface for the event notification.

evenCriteria : in TpUIEventCriteria

Specifies the new set of event criteria used by the application to define the event required. Only events that meet these criteria are reported.

*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA***Method***getNotification()**[This method is used by the application to query the event criteria set with createNotification or changeNotification.](#)[Returns: eventCriteria](#)[Specifies the event specific criteria used by the application to define the event required. Only events that meet these criteria are reported.](#)*Parameters*

No Parameters were identified for this method

Returns[TpUIEventCriteriaResultSet](#)*Raises***TpCommonExceptions, P_INVALID_CRITERIA**

8.2 Interface Class IpAppUIManager

Inherits from: IpInterface.

The Generic User Interaction Service manager application interface provides the application callback functions to the Generic User Interaction Service.

<<Interface>> IpAppUIManager
userInteractionAborted (userInteraction : in TpUIIdentifier) : void reportNotification (userInteraction : in TpUIIdentifier, eventInfo : in TpUIEventInfo, assignmentID : in TpAssignmentID) : IpAppUIRef userInteractionNotificationInterrupted () : void

[userInteractionNotificationContinued \(\) : void](#)

Method

userInteractionAborted()

This method indicates to the application that the User Interaction service instance has terminated or closed abnormally. No further communication will be possible between the User Interaction service instance and application.

Parameters

userInteraction : in TpUIIdentifier

Specifies the interface and sessionID of the user interaction service that has terminated.

Method

reportNotification()

[This method notifies the application of an occurred network event which matches the criteria installed by the createNotification method.](#)

[Returns: appUI](#)

[Specifies a reference to the application interface, which implements the callback interface for the new user interaction.](#)

Parameters

userInteraction : in TpUIIdentifier

Specifies the reference to the interface and the sessionID to which the notification relates.

eventInfo : in TpUIEventInfo

Specifies data associated with this event.

assignmentID : in TpAssignmentID

Specifies the assignment id which was returned by the createNotification() method. The application can use assignment id to associate events with event specific criteria and to act accordingly.

Returns

[IpAppUIRef](#)

Method

userInteractionNotificationInterrupted()

This method indicates to the application that all event notifications have been temporary interrupted (for example, due to faults detected). Note that more permanent failures are reported via the Framework (integrity management).

Parameters

No Parameters were identified for this method

*Method***userInteractionNotificationContinued()**

This method indicates to the application that event notifications will again be possible.

Parameters

No Parameters were identified for this method

8.3 Interface Class IpUI

Inherits from: IpService.

The User Interaction Service Interface provides functions to send information to, or gather information from the user. An application can use the User Interaction Service Interface independently of other services.

*Method***sendInfoReq()**

[This asynchronous method plays an announcement or sends other information to the user.](#)

[Returns: assignmentID](#)

[Specifies the ID assigned by the generic user interaction interface for a user interaction request.](#)

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the information to send to the user. This information can be:

- an infoID, identifying pre-defined information to be send (announcement and/or text);
- a string, defining the text to be sent;
- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal.

language : in TpLanguage

Specifies the Language of the information to be send to the user.

variableInfo : in TpUIVariableInfoSet

Defines the variable part of the information to send to the user.

repeatIndicator : in TpInt32

Defines how many times the information shall be sent to the end-user. A value of zero (0) indicates that the announcement shall be repeated until the call or call leg is released or an abortActionReq() is sent.

responseRequested : in TpUIResponseRequest

Specifies if a response is required from the call user interaction service, and any action the service should take.

Returns

[TpAssignmentID](#)

Raises

[TpCommonExceptions](#), [P_INVALID_SESSION_ID](#), [P_INVALID_NETWORK_STATE](#), [P_ILLEGAL_ID](#), [P_ID_NOT_FOUND](#)

*Method***sendInfoAndCollectReq()**

[This asynchronous method plays an announcement or sends other information to the user and collects some information from the user. The announcement usually prompts for a number of characters \(for example, these are digits or text strings such as "YES" if the user's terminal device is a phone\).](#)

[Returns: assignmentID](#)

[Specifies the ID assigned by the generic user interaction interface for a user interaction request.](#)

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the ID of the information to send to the user. This information can be:

- an infoID, identifying pre-defined information to be send (announcement and/or text);
- a string, defining the text to be sent;
- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal

language : in TpLanguage

Specifies the Language of the information to be send to the user.

variableInfo : in TpUIVariableInfoSet

Defines the variable part of the information to send to the user.

criteria : in TpUICollectCriteria

Specifies additional properties for the collection of information, such as the maximum and minimum number of characters, end character, first character timeout and inter-character timeout.

responseRequested : in TpUIResponseRequest

Specifies if a response is required from the call user interaction service, and any action the service should take. For this case it can especially be used to indicate e.g. the final request.

Returns

[TpAssignmentID](#)

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND, P_INVALID_CRITERIA, P_ILLEGAL_RANGE, P_INVALID_COLLECTION_CRITERIA

Method

release()

This method requests that the relationship between the application and the user interaction object be released. It causes the release of the used user interaction resources and interrupts any ongoing user interaction.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction created.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID

8.4 Interface Class IpAppUI

Inherits from: IpInterface.

The User Interaction Application Interface is implemented by the client application developer and is used to handle generic user interaction request responses and reports.

<<Interface>> IpAppUI
<p><u>sendInfoRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, response : in TpUIReport) : void</u></p> <p><u>sendInfoErr (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError) : void</u></p> <p><u>sendInfoAndCollectRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID,</u></p>

[response : in TpUIReport, collectedInfo : in TpString\) : void](#)

[sendInfoAndCollectErr \(userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError\) : void](#)

[userInteractionFaultDetected \(userInteractionSessionID : in TpSessionID, fault : in TpUIFault\) : void](#)

Method

sendInfoRes ()

This asynchronous method informs the application about the start or the completion of a sendInfoCallReq(). This response is called only if the responseRequested parameter of the sendInfoCallReq() method was set to P_UICALL_RESPONSE_REQUIRED.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the user.

Method

sendInfoErr ()

This asynchronous method indicates that the request to send information was unsuccessful.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

Method

sendInfoAndCollectRes ()

This asynchronous method returns the information collected to the application.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the user.

collectedInfo : in TpString

Specifies the information collected from the user.

*Method***sendInfoAndCollectErr()**

This asynchronous method indicates that the request to send information and collect a response was unsuccessful.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

*Method***userInteractionFaultDetected()**

This method indicates to the application that a fault has been detected in the user interaction.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the interface and sessionID of the user interaction service in which the fault has been detected.

fault : in TpUIFault

Specifies the fault that has been detected.

8.5 Interface Class IpUICall

Inherits from: IpUI.

The Call User Interaction Service Interface provides functions to send information to, or gather information from the user (or call party) to which a call leg is connected. An application can use the Call User Interaction Service Interface only in conjunction with another service interface, which provides mechanisms to connect a call leg to a user. At present, only the Call Control service supports this capability.

<<Interface>>

IpUICall

[recordMessageReq \(userInteractionSessionID : in TpSessionID, info : in TpUIInfo, criteria : in TpUIMessageCriteria\) : TpAssignmentID](#)
[deleteMessageReq \(usrInteractionSessionID : in TpSessionID, messageID : in TpInt32\) : TpAssignmentID](#)
[abortActionReq \(userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID\) : void](#)

*Method***recordMessageReq()**

[This asynchronous method allows the recording of a message. The recorded message can be played back at a later time with the sendInfoReq\(\) method.](#)

[Returns: assignmentID](#)

[Specifies the ID assigned by the generic user interaction interface for a user interaction request.](#)

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the information to send to the user. This information can be either an ID (for pre-defined announcement or text), a text string, or an URL (indicating the information to be sent, e.g. an audio stream).

criteria : in TpUIMessageCriteria

Defines the criteria for recording of messages

Returns

[TpAssignmentID](#)

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND, P_INVALID_CRITERIA

*Method***deleteMessageReq()**

[This asynchronous method allows to delete a recorded message.](#)

[Returns: assignmentID](#)

[Specifies the ID assigned by the generic user interaction interface for a user interaction request.](#)

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

messageID : in TpInt32

Specifies the message ID.

Returns

[TpAssignmentID](#)

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_ILLEGAL_ID, P_ID_NOT_FOUND

Method

abortActionReq()

This asynchronous method aborts a user interaction operation, e.g. a sendInfoReq(), from the specified call leg. The call and call leg are otherwise unaffected. The user interaction call service interrupts the current action on the specified leg.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the user interaction request to be cancelled.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_ASSIGNMENT_ID

8.6 Interface Class IpAppUICall

Inherits from: IpAppUI.

The Call User Interaction Application Interface is implemented by the client application developer and is used to handle call user interaction request responses and reports.

<<Interface>> IpAppUICall
<p><u>recordMessageRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, response : in TpUIReport, messageID : in TpInt32) : void</u></p> <p><u>recordMessageErr (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError) : void</u></p> <p><u>deleteMessageRes (usrInteractionSessionID : in TpSessionID, response : in TpUIReport, assignmentID : in TpAssignmentIDRef) : void</u></p> <p><u>deleteMessageErr (usrInteractionSessionID : in TpSessionID, error : in TpUIError, assignmentID : in TpAssignmentIDRef) : void</u></p>

[abortActionRes \(userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID\) : void](#)
[abortActionErr \(userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError\) : void](#)

Method

recordMessageRes ()

This method returns whether the message is successfully recorded or not. In case the message is recorded, the ID of the message is returned.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the device where the message is stored.

messageID : in TpInt32

Specifies the ID that was assigned to the message by the device where the message is stored.

Method

recordMessageErr ()

This method indicates that the request for recording of a message was not successful.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

Method

deleteMessageRes ()

This method returns whether the message is successfully deleted or not.

Parameters

usrInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

response : in TpUIReport

Specifies the type of response received from the device where the message was stored.

assignmentID : in TpAssignmentIDRef

Specifies the ID assigned by the call user interaction interface for a user interaction request.

*Method***deleteMessageErr()**

This method indicates that the request for deleting a message was not successful.

*Parameters***usrInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

error : in TpUIError

Specifies the error which led to the original request failing.

assignmentID : in TpAssignmentIDRef

Specifies the ID assigned by the call user interaction interface for a user interaction request.

*Method***abortActionRes()**

This asynchronous method confirms that the request to abort a user interaction operation on a call leg was successful.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

*Method***abortActionErr()**

This asynchronous method indicates that the request to abort a user interaction operation on a call leg resulted in an error.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

Error! No text of specified style in document.

19

Error! No text of specified style in document.

error : in TpUIError

Specifies the error which led to the original request failing.

=====Next changed section=====

B.7 All Interfaces

All methods on IpApp interfaces no longer throw exceptions.

All methods on the other interfaces throw TpCommonExceptions and individual, identified exceptions

[All methods now return void or the former out parameter.](#)

CHANGE REQUEST

⌘ **29.198-05 CR 004** ⌘ ev **-** ⌘ Current version: **4.2.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: ⌘ (U)SIM ME/UE Radio Access Network Core Network

Title:	⌘ Correction of description of sendInfoRes()		
Source:	⌘ CN5 *		
Work item code:	⌘ OSA1	Date:	⌘ 20/11/2001
Category:	⌘ F	Release:	⌘ Rel-4
	<i>Use one of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		<i>Use one of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ A wrong method name used in the description of method sendInfoRes() and also the semantics of the method is not correct.		
Summary of change:	⌘ The two errors are : - sendInfoCallReq() is a not existing method. Meant is method sendInfoReq(). - sendInfoRes only informs the Application of the completion of sendInfoReq() and not the start (see also the 'report' parameter of sendInfoRes).		
Consequences if not approved:	⌘ Ambiguous specification.		

Clauses affected:	⌘ Chapter 8.4		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	
Other comments:	⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

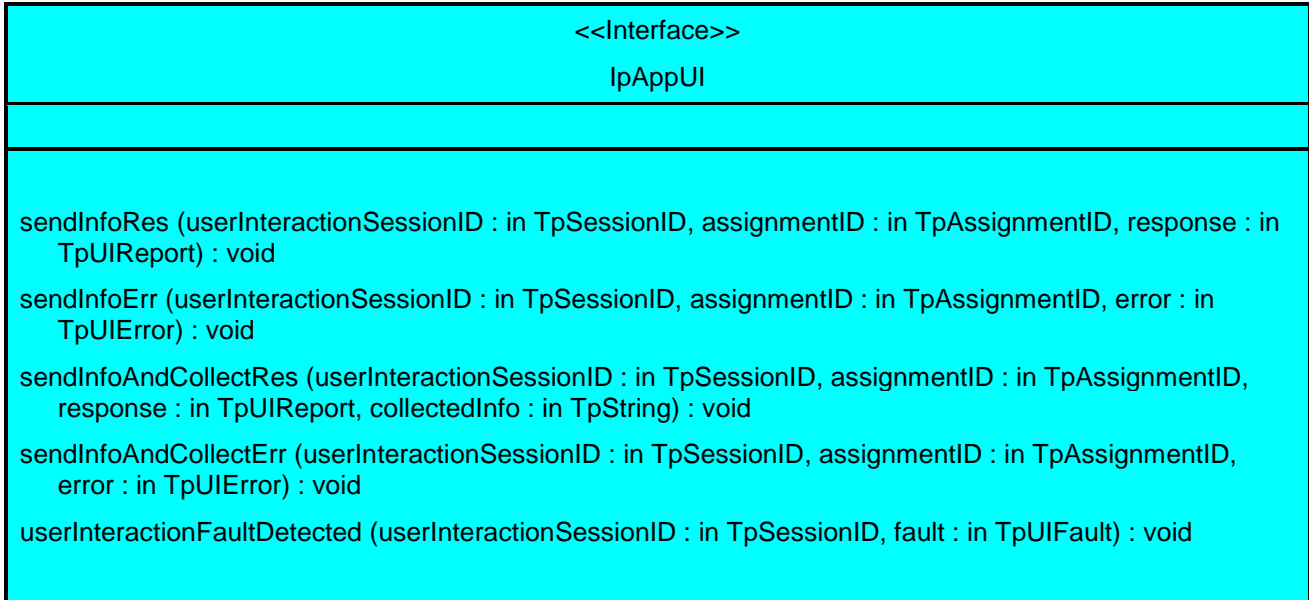
* Contact information: Corné Fonken, Ericsson Eurolab Netherlands, tel: +31 161 242639, e-mail: Corne.Fonken@eln.ericsson.se

- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

8.4 Interface Class IpAppUI

Inherits from: IpInterface.

The User Interaction Application Interface is implemented by the client application developer and is used to handle generic user interaction request responses and reports.



Method

sendInfoRes ()

This asynchronous method informs the application about the start or the completion of a sendInfoCallReq(). This response is called only if the responseRequested parameter of the sendInfoCallReq() method was set to P_UICALL_RESPONSE_REQUIRED.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the user.

CHANGE REQUEST

⌘ **29.198-05 CR 005** ⌘ rev **-** ⌘ Current version: **4.2.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: ⌘ (U)SIM ME/UE Radio Access Network Core Network

Title:	⌘ Correction to handling of deassign on related object		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 30/11/2001
Category:	⌘ F	Release:	⌘ REL-4
	<i>Use <u>one</u> of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900 .		<i>Use <u>one</u> of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ The application can indicate that the UICall object can be released after the final announcement is completed (i.e., by means of P_UI_FINAL_REQUEST) . When the application deassigns the related (call or callLeg) object the announcements will continue. However, now the application has to explicitly release the UICall object, even though it already indicated that the announcement resource could be released after the last announcement. E.g., imagine an application only wants to play some announcements and nothing else. Such an application would not be interested in the call and will therefore deassign the call after the announcements are requested. It would do: createUICall, sendInfoReq (FINAL), deassign on call. After this it still has to invoke lpUICall.release() according to the current spec.
Summary of change:	⌘ When the final announcement is already requested (i.e., in state release pending) a deassign should cause the UICall object to be destructed but the announcements should still continue. In the above example, the lpUICall.release() would not be needed.
Consequences if not approved:	⌘ The application has to request release of the UICall twice; once with final announcement and once explicitly after the deassign.

Clauses affected:	⌘ 9.3		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications	⌘	
	<input type="checkbox"/> Test specifications		
	<input type="checkbox"/> O&M Specifications		
Other comments:	⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.
- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.3 State Transition Diagrams for IpUICall

The state transition diagram shows the application view on the Call User Interaction object.

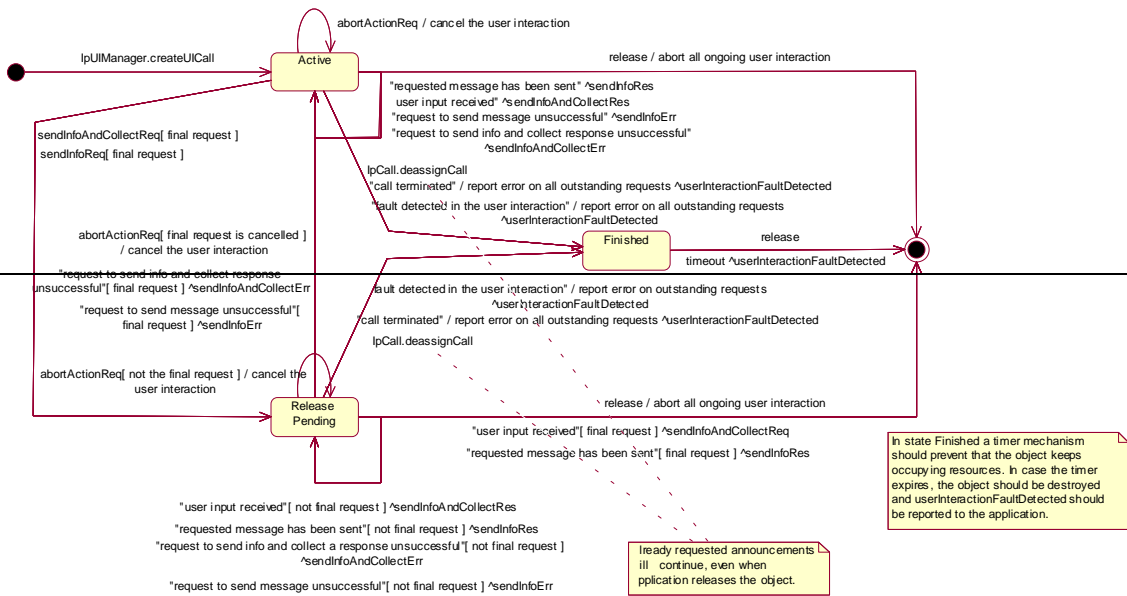
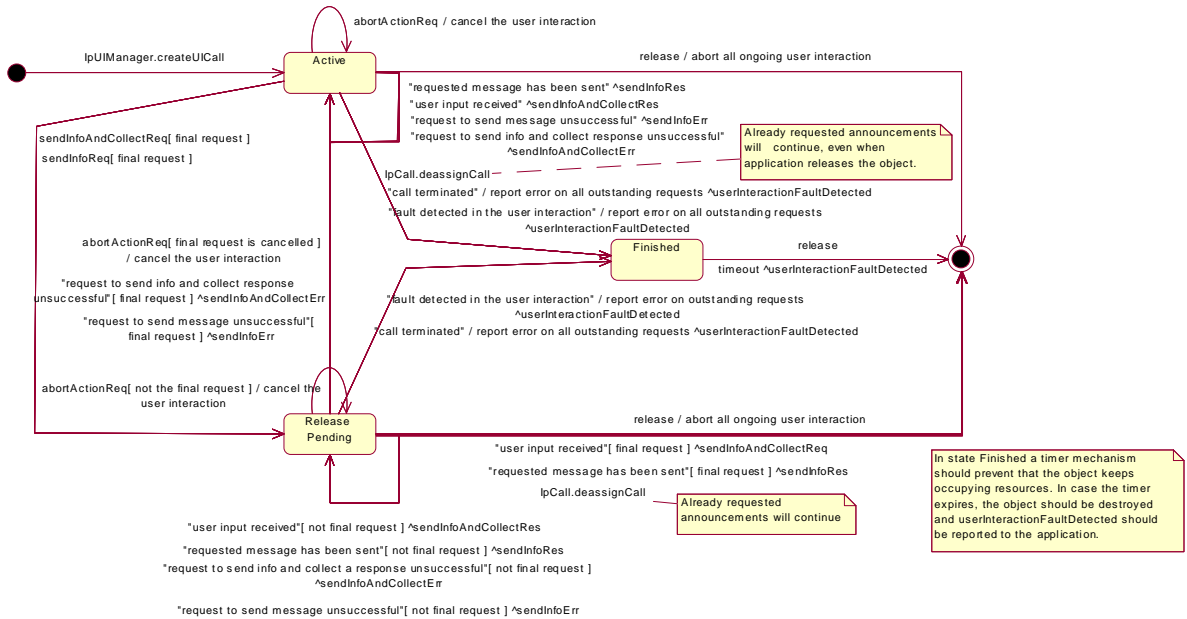


Figure : Application view on the UICall object

9.3.1 Active State

In this state a UICall object is available for announcements to be played to an end-user or obtaining information from the end-user.

When the application de-assigns the related Call or CallLeg object, a transition is made to the Finished state. However, all requested announcements will continue, even when the application releases the UICall object.

When the related call or callLeg is due to some reason terminated, a transition is made to the Finished state, the operation `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

9.3.2 Release Pending State

A transition to this state is made when the Application has indicated that after a certain announcement no further announcements need to be played to the end-user. There are, however, still a number of announcements that are not yet completed. When the last announcement is played or when the last user interaction has been obtained, the `UICall` object is destroyed. In case the final request failed or the application requested to abort the final request, a transition is made back to the Active state.

When the application de-assigns the related Call or CallLeg object, ~~a transition is made to the Finished state~~ the UICall object is destroyed. However, all requested announcements will continue, ~~even when the application releases the UICall object~~.

When the related call or callLeg is due to some reason terminated, a transition is made to the Finished state, the operation `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

9.3.3 Finished State

In this state the user interaction has ended. The application can only release the `UICall` object. Note that the application has to release the object itself as good OO practice requires that when an object is created on behalf of a certain entity, this entity is also responsible for destroying it when the object is no longer needed.

CHANGE REQUEST

⌘ **29.198-05 CR 006** ⌘ rev **-** ⌘ Current version: **4.2.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: ⌘ (U)SIM ME/UE Radio Access Network Core Network

Title:	⌘ Correction to Exceptions Raised in UI		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 30/11/2001
Category:	⌘ F	Release:	⌘ REL-4
	<i>Use <u>one</u> of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900.		<i>Use <u>one</u> of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ Essential Correction: Methods accepting an interface as a parameter need to be able to raise P_INVALID_INTERFACE_TYPE. changeNotification() parameter name is wrong.
Summary of change:	⌘ Added exception P_INVALID_INTERFACE_TYPE to createUI(), createUICall(), createNotification(). Corrected changeNotification() parameter evenCriteria to eventCriteria. Removed Tp..Ref and Ip..RefRef types which are misleading and not used. Various editorial spelling errors corrected.
Consequences if not approved:	⌘ Correct implementations of UI will not be possible as the method descriptions will not be complete. Some implementors might spot these errors and correct for them, others will not, leading to interworking problems. Parlay and ETSI have already corrected these errors - they need to be corrected in 29.198 to ensure synchronisation between groups and allow development of common applications across 3 platforms

Clauses affected:	⌘ 8.1, 8.2, 8.6, 11, B.10		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications	⌘	
	<input type="checkbox"/> Test specifications		
	<input type="checkbox"/> O&M Specifications		
Other comments:	⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be

downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

8 Generic User Interaction Interface Classes

The Generic User Interaction Service interface (GUI) is used by applications to interact with end users. The GUI is represented by the IpUIManager, IpUI and IpUICall interfaces that interface to services provided by the network. To handle responses and reports, the developer must implement IpAppUIManager and IpAppUI interfaces to provide the callback mechanism.

8.1 Interface Class IpUIManager

Inherits from: IpService.

This interface is the 'service manager' interface for the Generic User Interaction Service and provides the management functions to the Generic User Interaction Service.

<<Interface>> IpUIManager
<pre> createUI (appUI : in IpAppUIRef, userAddress : in TpAddress) : TpUIIdentifier createUICall (appUI : in IpAppUICallRef, uiTargetObject : in TpUITargetObject) : TpUICallIdentifier createNotification (appUIManager : in IpAppUIManagerRef, eventCriteria : in TpUIEventCriteria) : TpAssignmentID destroyNotification (assignmentID : in TpAssignmentID) : void changeNotification (assignmentID : in TpAssignmentID, eventCriteria : in TpUIEventCriteria) : void getNotification () : TpUIEventCriteriaResultSet </pre>

Method

createUI()

This method is used to create a new user interaction object for non-call related purposes

Results: userInteraction

Specifies the interface and sessionID of the user interaction created.

Parameters

appUI : in IpAppUIRef

Specifies the application interface for callbacks from the user interaction created.

userAddress : in TpAddress

Indicates the end-user with whom to interact.

*Returns***TpUIIdentifier***Raises***TpCommonExceptions, P_INVALID_NETWORK_STATE, P_INVALID_INTERFACE_TYPE***Method***createUICall()**

This method is used to create a new user interaction object for call related purposes.

The user interaction can take place to the specified party or to all parties in a call. Note that for certain implementation user interaction can only be performed towards the controlling call party, which shall be the only party in the call.

Returns: userInteraction

Specifies the interface and sessionID of the user interaction created.

*Parameters***appUI : in IpAppUICallRef**

Specifies the application interface for callbacks from the user interaction created.

uiTargetObject : in TpUITargetObject

Specifies the object on which to perform the user interaction. This can either be a Call, Multi-party Call or call leg object.

*Returns***TpUICallIdentifier***Raises***TpCommonExceptions, P_INVALID_NETWORK_STATE, P_INVALID_INTERFACE_TYPE***Method***createNotification()**

This method is used by the application to install specified notification criteria, for which the reporting is implicitly activated. If some application already requested notifications with criteria that overlap the specified criteria, the request is refused with P_INVALID_CRITERIA.

The criteria are said to overlap if both originating and terminating ranges overlap and the same number plan is used and the same servicecode is used.

If the same application requests two notifications with exactly the same criteria but different callback references, the second callback will be treated as an additional callback. This means that the callback will only be used in case when the first callback specified by the application is unable to handle the reportNotification (e.g., due to overload or failure).

Returns: assignmentID

Specifies the ID assigned by the generic user interaction manager interface for this newly installed notification criteria.

*Parameters***appUIManager : in IpAppUIManagerRef**

If this parameter is set (i.e. not NULL) it specifies a reference to the application interface, which is used for callbacks. If set to NULL, the application interface defaults to the interface specified via the setCallback() method.

eventCriteria : in TpUIEventCriteria

Specifies the event specific criteria used by the application to define the event required, like user address and service code.

*Returns***TpAssignmentID***Raises***TpCommonExceptions, P_INVALID_CRITERIA, P_INVALID_INTERFACE_TYPE***Method***destroyNotification()**

This method is used by the application to destroy previously installed notification criteria via the createNotification method.

*Parameters***assignmentID : in TpAssignmentID**

Specifies the assignment ID given by the generic user interaction manager interface when the previous createNotification() was called. If the assignment ID does not correspond to one of the valid assignment IDs, the framework will return the error code P_INVALID_ASSIGNMENT_ID.

*Raises***TpCommonExceptions, P_INVALID_ASSIGNMENT_ID***Method***changeNotification()**

This method is used by the application to change the event criteria introduced with createNotification method. Any stored notification request associated with the specified assignmentID will be replaced with the specified events requested.

*Parameters***assignmentID : in TpAssignmentID**

Specifies the ID assigned by the manager interface for the event notification.

eventCriteria : in TpUIEventCriteria

Specifies the new set of event criteria used by the application to define the event required. Only events that meet these criteria are reported.

Raises

TpCommonExceptions, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA

Method

getNotification()

This method is used by the application to query the event criteria set with createNotification or changeNotification.

Returns: eventCriteria

Specifies the event specific criteria used by the application to define the event required. Only events that meet these criteria are reported.

Parameters

No Parameters were identified for this method

Returns

TpUIEventCriteriaResultSet

Raises

TpCommonExceptions, P_INVALID_CRITERIA

8.2 Interface Class IpAppUIManager

Inherits from: IpInterface.

The Generic User Interaction Service manager application interface provides the application callback functions to the Generic User Interaction Service.

<<Interface>> IpAppUIManager
userInteractionAborted (userInteraction : in TpUIIdentifier) : void reportNotification (userInteraction : in TpUIIdentifier, eventInfo : in TpUIEventInfo, assignmentID : in TpAssignmentID) : IpAppUIRef userInteractionNotificationInterrupted () : void userInteractionNotificationContinued () : void

Method

userInteractionAborted()

This method indicates to the application that the User Interaction service instance has terminated or closed abnormally. No further communication will be possible between the User Interaction service instance and application.

*Parameters***userInteraction : in TpUIIdentifier**

Specifies the interface and sessionID of the user interaction service that has terminated.

*Method***reportNotification()**

This method notifies the application of an occurred network event which matches the criteria installed by the createNotification method.

Returns: appUI

Specifies a reference to the application interface, which implements the callback interface for the new user interaction.

*Parameters***userInteraction : in TpUIIdentifier**

Specifies the reference to the interface and the sessionID to which the notification relates.

eventInfo : in TpUIEventInfo

Specifies data associated with this event.

assignmentID : in TpAssignmentID

Specifies the assignment id which was returned by the createNotification() method. The application can use assignment id to associate events with event specific criteria and to act accordingly.

*Returns***IpAppUIRef***Method***userInteractionNotificationInterrupted()**

This method indicates to the application that all event notifications have been temporarily interrupted (for example, due to faults detected). Note that more permanent failures are reported via the Framework (integrity management).

Parameters

No Parameters were identified for this method

*Method***userInteractionNotificationContinued()**

This method indicates to the application that event notifications will again be possible.

Parameters

No Parameters were identified for this method

8.3 Interface Class IpUI

Inherits from: IpService.

The User Interaction Service Interface provides functions to send information to, or gather information from the user. An application can use the User Interaction Service Interface independently of other services.

<<Interface>> IpUI
<pre> sendInfoReq (userInteractionSessionID : in TpSessionID, info : in TpUIInfo, language : in TpLanguage, variableInfo : in TpUIVariableInfoSet, repeatIndicator : in TpInt32, responseRequested : in TpUIResponseRequest) : TpAssignmentID sendInfoAndCollectReq (userInteractionSessionID : in TpSessionID, info : in TpUIInfo, language : in TpLanguage, variableInfo : in TpUIVariableInfoSet, criteria : in TpUICollectCriteria, responseRequested : in TpUIResponseRequest) : TpAssignmentID release (userInteractionSessionID : in TpSessionID) : void </pre>

Method

sendInfoReq()

This asynchronous method plays an announcement or sends other information to the user.

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the information to send to the user. This information can be:

- an infoID, identifying pre-defined information to be send (announcement and/or text);
- a string, defining the text to be sent;
- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal.

language : in TpLanguage

Specifies the Language of the information to be send to the user.

variableInfo : in TpUIVariableInfoSet

Defines the variable part of the information to send to the user.

repeatIndicator : in TpInt32

Defines how many times the information shall be sent to the end-user. A value of zero (0) indicates that the announcement shall be repeated until the call or call leg is released or an abortActionReq() is sent.

responseRequested : in TpUIResponseRequest

Specifies if a response is required from the call user interaction service, and any action the service should take.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND

*Method***sendInfoAndCollectReq()**

This asynchronous method plays an announcement or sends other information to the user and collects some information from the user. The announcement usually prompts for a number of characters (for example, these are digits or text strings such as "YES" if the user's terminal device is a phone).

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the ID of the information to send to the user. This information can be:

- an infoID, identifying pre-defined information to be send (announcement and/or text);
- a string, defining the text to be sent;
- a URL , identifying pre-defined information or data to be sent to or downloaded into the terminal

language : in TpLanguage

Specifies the Language of the information to be send to the user.

variableInfo : in TpUIVariableInfoSet

Defines the variable part of the information to send to the user.

criteria : in TpUICollectCriteria

Specifies additional properties for the collection of information, such as the maximum and minimum number of characters, end character, first character timeout and inter-character timeout.

responseRequested : in TpUIResponseRequest

Specifies if a response is required from the call user interaction service, and any action the service should take. For this case it can especially be used to indicate e.g. the final request.

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND, P_INVALID_CRITERIA, P_ILLEGAL_RANGE, P_INVALID_COLLECTION_CRITERIA

Method

release()

This method requests that the relationship between the application and the user interaction object be released. It causes the release of the used user interaction resources and interrupts any ongoing user interaction.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction created.

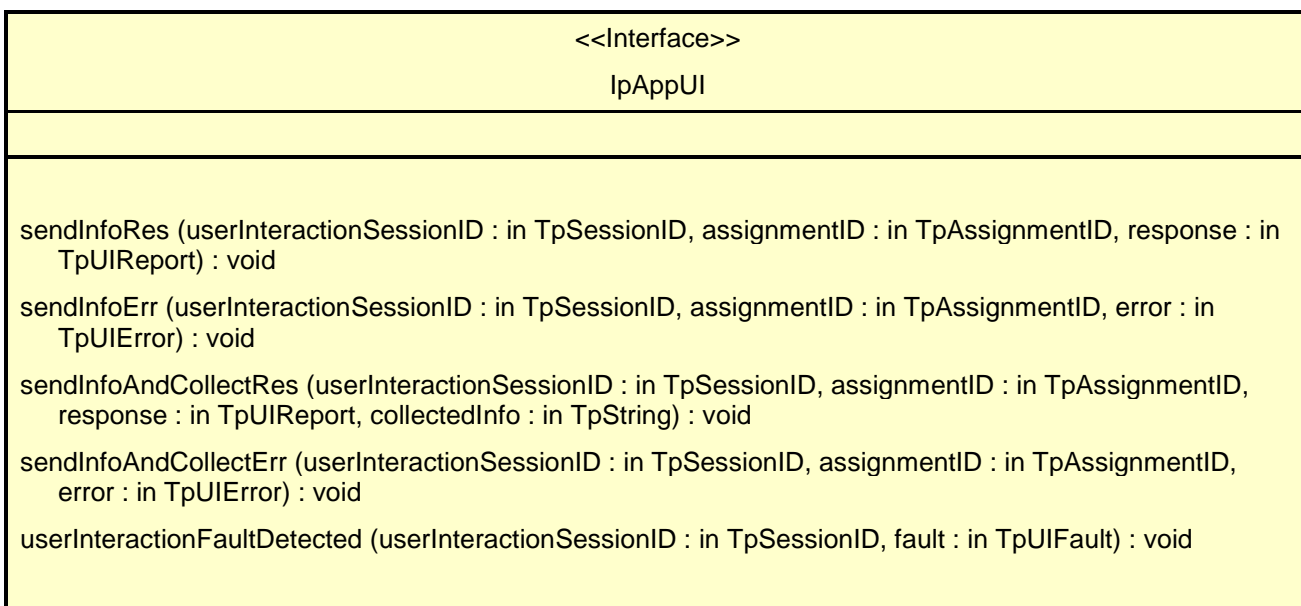
Raises

TpCommonExceptions, P_INVALID_SESSION_ID

8.4 Interface Class IpAppUI

Inherits from: IpInterface.

The User Interaction Application Interface is implemented by the client application developer and is used to handle generic user interaction request responses and reports.



*Method***sendInfoRes ()**

This asynchronous method informs the application about the start or the completion of a sendInfoCallReq(). This response is called only if the responseRequested parameter of the sendInfoCallReq() method was set to P_UICALL_RESPONSE_REQUIRED.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the user.

*Method***sendInfoErr ()**

This asynchronous method indicates that the request to send information was unsuccessful.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

*Method***sendInfoAndCollectRes ()**

This asynchronous method returns the information collected to the application.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the user.

collectedInfo : in TpString

Specifies the information collected from the user.

*Method***sendInfoAndCollectErr ()**

This asynchronous method indicates that the request to send information and collect a response was unsuccessful.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

*Method***userInteractionFaultDetected ()**

This method indicates to the application that a fault has been detected in the user interaction.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the interface and sessionID of the user interaction service in which the fault has been detected.

fault : in TpUIFault

Specifies the fault that has been detected.

8.5 Interface Class IpUICall

Inherits from: IpUI.

The Call User Interaction Service Interface provides functions to send information to, or gather information from the user (or call party) to which a call leg is connected. An application can use the Call User Interaction Service Interface only in conjunction with another service interface, which provides mechanisms to connect a call leg to a user. At present, only the Call Control service supports this capability.

<<Interface>> IpUICall
recordMessageReq (userInteractionSessionID : in TpSessionID, info : in TpUIInfo, criteria : in TpUIMessageCriteria) : TpAssignmentID deleteMessageReq (usrInteractionSessionID : in TpSessionID, messageID : in TpInt32) : TpAssignmentID abortActionReq (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID) : void

*Method***recordMessageReq()**

This asynchronous method allows the recording of a message. The recorded message can be played back at a later time with the sendInfoReq() method.

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

info : in TpUIInfo

Specifies the information to send to the user. This information can be either an ID (for pre-defined announcement or text), a text string, or an URL (indicating the information to be sent, e.g. an audio stream).

criteria : in TpUIMessageCriteria

Defines the criteria for recording of messages

*Returns***TpAssignmentID***Raises*

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE, P_ILLEGAL_ID, P_ID_NOT_FOUND, P_INVALID_CRITERIA

*Method***deleteMessageReq()**

This asynchronous method allows to delete a recorded message.

Returns: assignmentID

Specifies the ID assigned by the generic user interaction interface for a user interaction request.

*Parameters***usrInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

messageID : in TpInt32

Specifies the message ID.

*Returns***TpAssignmentID***Raises***TpCommonExceptions, P_INVALID_SESSION_ID, P_ILLEGAL_ID, P_ID_NOT_FOUND***Method***abortActionReq()**

This asynchronous method aborts a user interaction operation, e.g. a sendInfoReq(), from the specified call leg. The call and call leg are otherwise unaffected. The user interaction call service interrupts the current action on the specified leg.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the user interaction request to be cancelled.

*Raises***TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_ASSIGNMENT_ID**

8.6 Interface Class IpAppUICall

Inherits from: IpAppUI.

The Call User Interaction Application Interface is implemented by the client application developer and is used to handle call user interaction request responses and reports.

<<Interface>> IpAppUICall
recordMessageRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, response : in TpUIReport, messageID : in TpInt32) : void recordMessageErr (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError) : void <u>deleteMessageRes (usrInteractionSessionID : in TpSessionID, response : in TpUIReport, assignmentID : in TpAssignmentID) : void</u> <u>deleteMessageErr (usrInteractionSessionID : in TpSessionID, error : in TpUIError, assignmentID : in TpAssignmentID) : void</u> abortActionRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID) : void abortActionErr (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, error : in TpUIError) : void

*Method***recordMessageRes ()**

This method returns whether the message is successfully recorded or not. In case the message is recorded, the ID of the message is returned.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

response : in TpUIReport

Specifies the type of response received from the device where the message is stored.

messageID : in TpInt32

Specifies the ID that was assigned to the message by the device where the message is stored.

*Method***recordMessageErr ()**

This method indicates that the request for recording of a message was not successful.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

*Method***deleteMessageRes()**

This method returns whether the message is successfully deleted or not.

*Parameters***usrInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

response : in TpUIReport

Specifies the type of response received from the device where the message was stored.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

*Method***deleteMessageErr()**

This method indicates that the request for deleting a message was not successful.

*Parameters***usrInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

error : in TpUIError

Specifies the error which led to the original request failing.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

*Method***abortActionRes()**

This asynchronous method confirms that the request to abort a user interaction operation on a call leg was successful.

*Parameters***userInteractionSessionID : in TpSessionID**

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

Method

abortActionErr()

This asynchronous method indicates that the request to abort a user interaction operation on a call leg resulted in an error.

Parameters

userInteractionSessionID : in TpSessionID

Specifies the user interaction session ID of the user interaction.

assignmentID : in TpAssignmentID

Specifies the ID assigned by the call user interaction interface for a user interaction request.

error : in TpUIError

Specifies the error which led to the original request failing.

9 State Transition Diagrams

9.1 State Transition Diagrams for IpUIManager

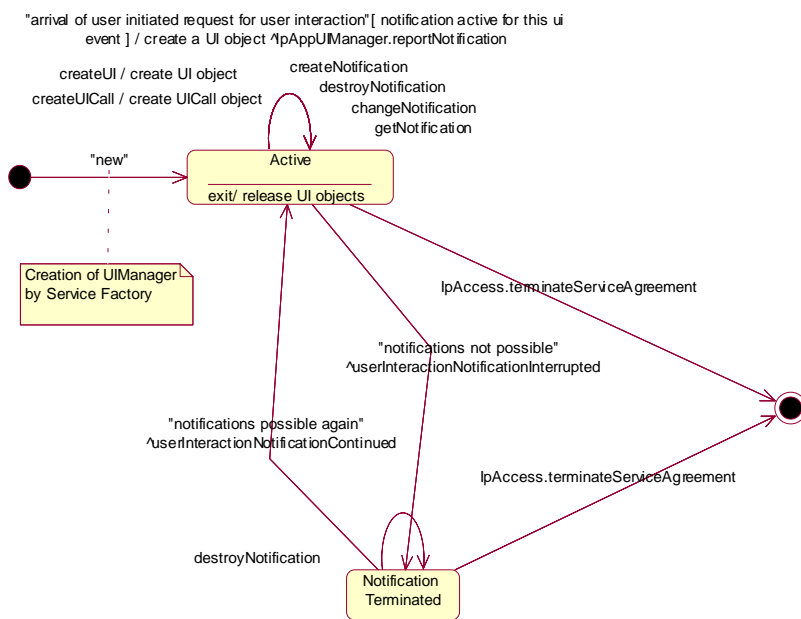


Figure : Application view on the UI Manager

9.1.1 Active State

In this state a relation between the Application and a User Interaction Service Capability Feature (Generic User Interaction or Call User Interaction) has been established. The application is now able to request creation of UI and/orUICall objects.

9.1.2 Notification Terminated State

When the UI manager is in the Notification terminated state, events requested with createNotification() will not be forwarded to the application. There can be multiple reasons for this: for instance it might be that the application receives more notifications than defined in the Service Level Agreement. Another example is that the SCS has detected it receives no notifications from the network due to e.g. a link failure. In this state no requests for new notifications will be accepted.

9.2 State Transition Diagrams for IpUI

The state transition diagram shows the application view on the User Interaction object.

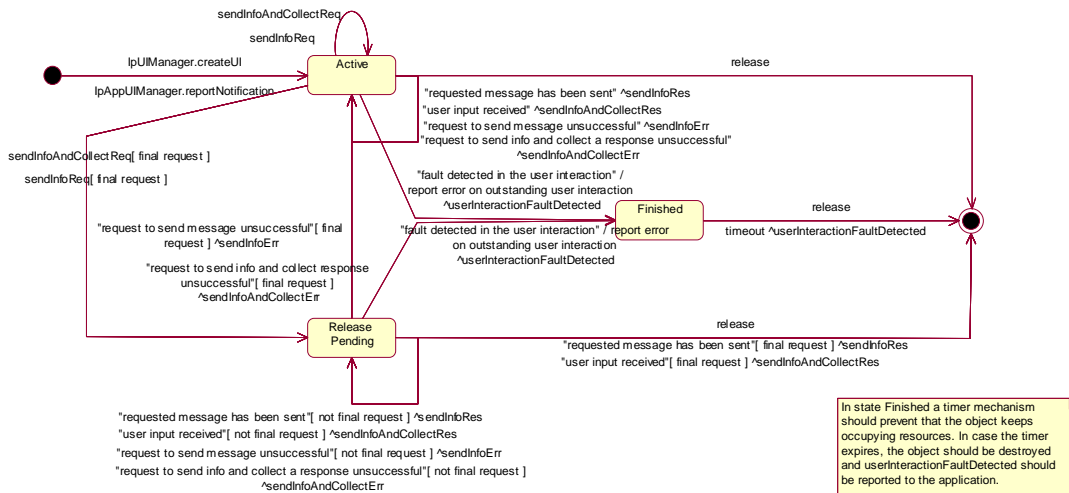


Figure : Application view on the UI object

9.2.1 Active State

In this state the UI object is available for requesting messages to be send to the network.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), userInteractionFaultDetected() will be invoked on the application and an error will be reported on all outstanding requests.

9.2.2 Release Pending State

A transition to this state is made when the Application has indicated that after a certain message no further messages need to be sent to the end-user. There are, however, still a number of messages that are not yet completed. When the last message is sent or when the last user interaction has been obtained, the UI object is destroyed.

In case the final request failed or the application requested to abort the final request, a transition is made back to the Active state.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), userInteractionFaultDetected() will be invoked on the application and an error will be reported on all outstanding requests.

9.2.3 Finished State

In this state the user interaction has ended. The application can only release the UI object. Note that the application has to release the object itself as good OO practice requires that when an object is created on behalf of a certain entity, this entity is also responsible for destroying it when the object is no longer needed.

9.3 State Transition Diagrams for IpUICall

The state transition diagram shows the application view on the Call User Interaction object.

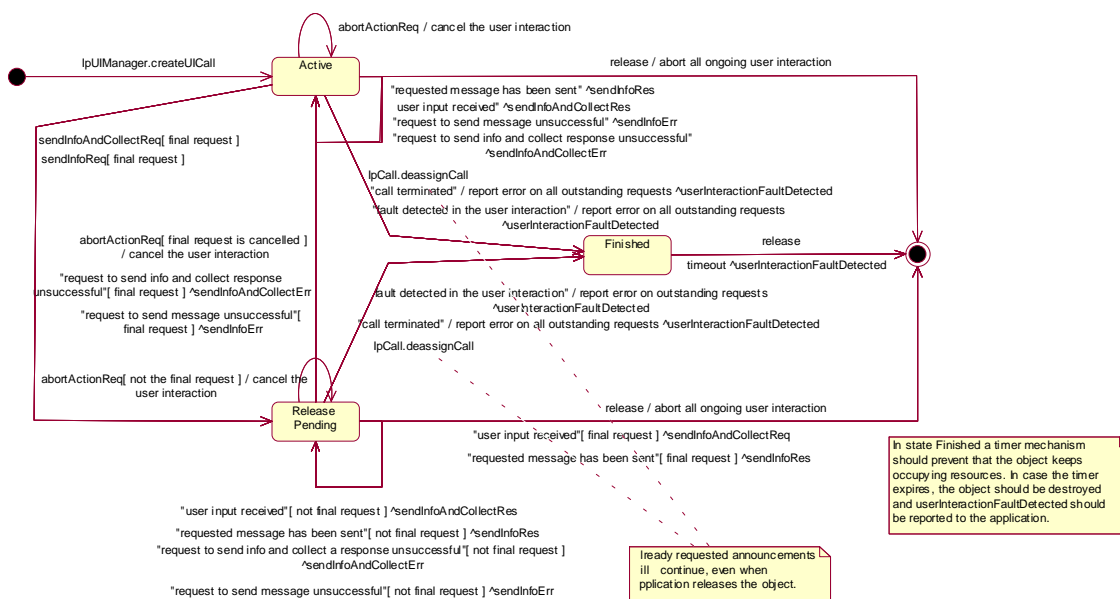


Figure : Application view on the UICall object

9.3.1 Active State

In this state a UICall object is available for announcements to be played to an end-user or obtaining information from the end-user.

When the application de-assigns the related Call object, a transition is made to the Finished state. However, all requested announcements will continue, even when the application releases the UICall object.

When the related call is due to some reason terminated, a transition is made to the Finished state, the operation userInteractionFaultDetected() will be invoked on the application and an error will be reported on all outstanding requests.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

9.3.2 Release Pending State

A transition to this state is made when the Application has indicated that after a certain announcement no further announcements need to be played to the end-user. There are, however, still a number of announcements that are not yet completed. When the last announcement is played or when the last user interaction has been obtained, the `UICall` object is destroyed. In case the final request failed or the application requested to abort the final request, a transition is made back to the Active state.

When the application de-assigns the related `Call` object, a transition is made to the Finished state. However, all requested announcements will continue, even when the application releases the `UICall` object.

When the related call is due to some reason terminated, a transition is made to the Finished state, the operation `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

In case a fault is detected on the user interaction (e.g. a link failure to the IVR system), `userInteractionFaultDetected()` will be invoked on the application and an error will be reported on all outstanding requests.

9.3.3 Finished State

In this state the user interaction has ended. The application can only release the `UICall` object. Note that the application has to release the object itself as good OO practice requires that when an object is created on behalf of a certain entity, this entity is also responsible for destroying it when the object is no longer needed.

10 Service Properties

10.1 User Interaction Service Properties

The following table lists properties relevant for the User Interaction API.

Property	Type	Description
P_INFO_TYPE	INTEGER_SET	Specifies whether the UI SCS supports text or URLs etc. Allowed value set: {P_INFO_ID, P_URL, P_TEXT}

The previous table lists properties related to capabilities of the SCS itself. The following table lists properties that are used in the context of the Service Level Agreement, e.g. to restrict the access of applications to the capabilities of the SCS.

Property	Type	Description
P_TRIGGERING_ADDRESSES	ADDRESS_RANGE_SET	Specifies which numbers the notification may be set
P_SERVICE_CODE	INTEGER_SET	Specifies the service codes that may be used for notification requests.

11 Data Definitions

11.1 TpUIFault

Defines the cause of the UI fault detected.

Name	Value	Description
P_UI_FAULT_UNDEFINED	0	Undefined
P_UI_CALL_ENDED	1	The related Call object has been terminated. Therefore, the UICall object is also terminated. No further interaction is possible with this object.

11.2 IpUI

Defines the address of an IpUI Interface.

11.3 IpUIRef

Defines a Reference to type IpUI.

~~11.4 IpUIRefRef~~

~~Defines a Reference to type IpUIRef.~~

~~11.5~~ 11.4 IpAppUI

Defines the address of an IpAppUI Interface.

~~11.6~~ 11.5 IpAppUIRef

Defines a Reference to type IpAppUI.

~~11.7 IpAppUIRefRef~~

~~Defines a Reference to type IpAppUIRef.~~

~~11.8~~ 11.6 IpAppUIManager

Defines the address of an IpAppUIManager Interface.

~~11.9~~ 11.7 IpAppUIManagerRef

Defines a Reference to type IpAppUIManager .

11.10 11.8 TpUICallIdentifier

Defines the Sequence of Data Elements that unambiguously specify the UICall object.

Structure Element Name	Structure Element Type	Structure Element Description
UICallRef	IpUICallRef	This element specifies the interface reference for the UICall object.
UserInteractionSessionID	TpSessionID	This element specifies the User Interaction session ID.

11.11 TpUICallIdentifierRef

Defines a reference to type TpUICallIdentifier.

11.12 11.9 TpUICollectCriteria

Defines the Sequence of Data Elements that specify the additional properties for the collection of information, such as the end character, first character timeout, inter-character timeout, and maximum interaction time.

Structure Element Name	Structure Element Type
MinLength	TpInt32
MaxLength	TpInt32
EndSequence	TpString
StartTimeout	TpDuration
InterCharTimeout	TpDuration

The structure elements specify the following criteria:

MinLength: Defines the minimum number of characters (e.g. digits) to collect.

MaxLength: Defines the maximum number of characters (e.g. digits) to collect.

EndSequence: Defines the character or characters which terminate an input of variable length, e.g. phonenumbers.

StartTimeout: specifies the value for the first character time-out timer. The timer is started when the announcement has been completed or has been interrupted. The user should enter the start of the response (e.g. first digit) before the timer expires. If the start of the response is not entered before the timer expires, the input is regarded to be erroneous. After receipt of the start of the response, which may be valid or invalid, the timer is stopped.

InterCharTimeOut: specifies the value for the inter-character time-out timer. The timer is started when a response (e.g. digit) is received, and is reset and restarted when a subsequent response is received. The responses may be valid or invalid. the announcement has been completed or has been interrupted.

Input is considered successful if the following applies:

If the EndSequence is not present (i.e. NULL):

- when the InterCharTimeOut timer expires; or
- when the number of valid digits received equals the MaxLength.

If the EndSequence is present:

- when the InterCharTimeOut timer expires; or
- when the EndSequence is received; or
- when the number of valid digits received equals the MaxLength.

In the case the number of valid characters received is less than the `MinLength` when the `InterCharTimeOut` timer expires or when the `EndSequence` is received, the input is considered erroneous.

The collected characters (including the `EndSequence`) are sent to the client application when input has been successful.

41.1311.10 TpUIError

Defines the UI error codes.

Name	Value	Description
P_UI_ERROR_UNDEFINED	0	Undefined error
P_UI_ERROR_ILLEGAL_INFO	1	The specified information (InfoId, InfoData, or InfoAddress) is invalid
P_UI_ERROR_ID_NOT_FOUND	2	A legal InfoId is not known to the the User Interaction service
P_UI_ERROR_RESOURCE_UNAVAILABLE	3	The information resources used by the User Interaction service are unavailable, e.g. due to an overload situation.
P_UI_ERROR_ILLEGAL_RANGE	4	The values for minimum and maximum collection length are out of range
P_UI_ERROR_IMPROPER_USER_RESPONSE	5	Improper user response
P_UI_ERROR_ABANDON	6	The specified leg is disconnected before the send information completed
P_UI_ERROR_NO_OPERATION_ACTIVE	7	There is no active User Interaction for the specified leg. Either the application did not start any User Interaction or the User Interaction was already finished when the <code>abortAction_Req()</code> was called.
P_UI_ERROR_NO_SPACE_AVAILABLE	8	There is no more storage capacity to record the message when the <code>recordMessage()</code> operation was called
P_UI_ERROR_RESOURCE_TIMEOUT	9	The request has been accepted by the resource but it did not report a result.

The call `User Interaction` object will be automatically de-assigned if the error `P_UI_ERROR_ABANDON` is reported, as a corresponding call or call leg object no longer exists.

41.1411.11 TpUIEventCriteria

Defines the Sequence of Data Elements that specify the additional criteria for receiving a UI notification

Structure Element Name	Structure Element Type	Description
OriginatingAddress	TpAddressRange	Defines the originating address for which the notification is requested.
DestinationAddress	TpAddressRange	Defines the destination address or address range for which the notification is requested.
ServiceCode	TpString	Defines a 2-digit code indicating the UI to be triggered. The value is operator specific.

41.15TpUIEventCriteriaResultSetRef

Defines a reference to `TpUIEventCriteriaResultSet`.

41.1611.12 TpPUIEventCriteriaResultSet

Defines a set of `TpUIEventCriteriaResult`.

41.1711.13 TpPUIEventCriteriaResult

Defines a sequence of data elements that specify a requested event notification criteria with the associated assignmentID.

Structure Element Name	Structure Element Type	Structure Element Description
EventCriteria	TpUIEventCriteria	The event criteria that were specified by the application.
AssignmentID	TpInt32	The associated assignmentID. This can be used to disable the notification.

41.1811.14 TpUIEventInfo

Defines the Sequence of Data Elements that specify a UI notification

Structure Element Name	Structure Element Type	Structure Element Description
OriginatingAddress	TpAddress	Defines the originating address.
DestinationAddress	TpAddress	Defines the destination address.
ServiceCode	TpString	Defines a 2-digit code indicating the UI to be triggered. The value is operator specific.
DataTypeIndication	TpUIEventInfoDataType	Identifies the type of contents in the dataString.
DataString	TpString	Freely defined data string with a limited length e.g. 160 bytes according to the network policy.

41.1911.15 TpUIEventInfoDataType

Defines the type of the dataString parameter in the method userInteractionEventNotify.

Name	Value	Description
P_UI_EVENT_DATA_TYPE_UNDEFINED	0	Undefined (e.g. binary data)
P_UI_EVENT_DATA_TYPE_UNSPECIFIED	1	Unspecified data
P_UI_EVENT_DATA_TYPE_TEXT	2	Text
P_UI_EVENT_DATA_TYPE_USSD_DATA	3	USSD data starting with coding scheme

41.2011.16 TpUIIdentifier

Defines the Sequence of Data Elements that unambiguously specify the UI object

Structure Element Name	Structure Element Type	Structure Element Description
UIRef	IpUIRef	This element specifies the interface reference for the UI object.
UserInteractionSessionID	TpSessionID	This element specifies the User Interaction session ID.

41.21TpUIIdentifierRef

Defines a reference to type TpUIIdentifier.

41.2211.17 TpUIInfo

Defines the Tagged Choice of Data Elements that specify the information to send to the user.

	Tag Element Type	
	TpUIInfoType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_INFO_ID	TpInt32	InfoId
P_UI_INFO_DATA	TpString	InfoData
P_UI_INFO_ADDRESS	TpURL	InfoAddress

The choice elements represents the following:

InfoID: defines the ID of the user information script or stream to send to an end-user. The values of this data type are operator specific.

InfoData: defines the data to be sent to an end-user’s terminal. The data is free-format and the encoding is depending on the resources being used..

InfoAddress: defines the URL of the text or stream to be sent to an end-user’s terminal.

41.2311.18 TpUIInfoType

Defines the type of the information to be send to the user.

Name	Value	Description
P_UI_INFO_ID	1	The information to be send to an end-user consists of an ID
P_UI_INFO_DATA	2	The information to be send to an end-user consists of a data string
P_UI_INFO_ADDRESS	3	The information to be send to an end-user consists of a URL.

41.2411.19 TpUIMessageCriteria

Defines the Sequence of Data Elements that specify the additional properties for the recording of a message.

Structure Element Name	Structure Element Type
EndSequence	TpString
MaxMessageTime	TpDuration
MaxMessageSize	TpInt32

The structure elements specify the following criteria:

EndSequence: Defines the character or characters which terminate an input of variable length, e.g. phonenumbers.

MaxMessageTime: specifies the maximum duration in seconds of the message that is to be recorded.

MaxMessageSize: If this parameter is non-zero, it specifies the maximum size in bytes of the message that is to be recorded.

41.2511.20 TpUIReport

Defines the UI reports if a response was requested.

Name	Value	Description
P_UI_REPORT_UNDEFINED	0	Undefined report
P_UI_REPORT_INFO_SENT	1	Confirmation that the information has been sent
P_UI_REPORT_INFO_COLLECTED	2	Information collected., meeting the specified criteria.
P_UI_REPORT_NO_INPUT	3	No information collected. The user immediately entered the delimiter character. No valid information has been returned
P_UI_REPORT_TIMEOUT	4	No information collected. The user did not input any response before the input timeout expired
P_UI_REPORT_MESSAGE_STORED	5	A message has been stored successfully
P_UI_REPORT_MESSAGE_NOT_STORED	6	The message has not been stored successfully
P_UI_REPORT_MESSAGE_DELETED	7	A message has been deleted successfully
P_UI_REPORT_MESSAGE_NOT_DELETED	8	A message has not been deleted successfully

41.2611.21 TpUIResponseRequest

Defines the situations for which a response is expected following the User Interaction.

Name	Value	Description
P_UI_RESPONSE_REQUIRED	1	The User Interaction Call shall send a response when the request has completed.
P_UI_LAST_ANNOUNCEMENT_IN_A_ROW	2	This is the final announcement within a sequence. It might, however, be that additional announcements will be requested at a later moment. The User Interaction Call service may release any used resources in the network. The UI object will not be released.
P_UI_FINAL_REQUEST	4	This is the final request. The UI object will be released after the information has been presented to the user.

This parameter represent a so-called bitmask, i.e. the values can be added to derived the final meaning.

41.2711.22 TpUITargetObjectType

Defines the type of object where User Interaction should be performed upon.

Name	Value	Description
P_UI_TARGET_OBJECT_CALL	0	User-interaction will be performed on a complete Call.
P_UI_TARGET_OBJECT_MULTI_PARTY_CALL	1	User-interaction will be performed on a complete Multi-party Call.
P_UI_TARGET_OBJECT_CALL_LEG	2	User-interaction will be performed on a single Call Leg.

41.2811.23 TpUITargetObject

Defines the Tagged Choice of Data Elements that specify the object to perform User Interaction on.

Tag Element Type		
	TpUITargetObjectType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_TARGET_OBJECT_CALL	TpCallIdentifier	Call
P_UI_TARGET_OBJECT_MULTI_PARTY_CALL	TpMultiPartyCallIdentifier	MultiPartyCall
P_UI_TARGET_OBJECT_CALL_LEG	TpCallLegIdentifier	CallLeg

41.2911.24 TpUIVariableInfo

Defines the Tagged Choice of Data Elements that specify the variable parts in the information to send to the user.

Tag Element Type		
	TpUIVariablePartType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_VARIABLE_PART_INT	TpInt32	VariablePartInteger
P_UI_VARIABLE_PART_ADDRESS	TpString	VariablePartAddress
P_UI_VARIABLE_PART_TIME	TpTime	VariablePartTime
P_UI_VARIABLE_PART_DATE	TpDate	VariablePartDate
P_UI_VARIABLE_PART_PRICE	TpPrice	VariablePartPrice

41.3011.25 TpUIVariableInfoSet

Defines a Numbered Set of Data Elements of TpUIVariableInfo.

41.3111.26 TpUIVariablePartType

Defines the type of the variable parts in the information to send to the user.

Name	Value	Description
P_UI_VARIABLE_PART_INT	0	Variable part is of type integer
P_UI_VARIABLE_PART_ADDRESS	1	Variable part is of type address
P_UI_VARIABLE_PART_TIME	2	Variable part is of type time
P_UI_VARIABLE_PART_DATE	3	Variable part is of type date
P_UI_VARIABLE_PART_PRICE	4	Variable part is of type price

12 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_ILLEGAL_ID	Information id specified is invalid
P_ID_NOT_FOUND	A legal information id is not known to the User Interaction Service
P_ILLEGAL_RANGE	The values for minimum and maximum collection length are out of range.
P_INVALID_COLLECTION_CRITERIA	Invalid collection criteria specified

Each exception class contains the following structure:

Structure Element Name	Structure Element Type	Structure Element Description
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name

Annex A (normative): OMG IDL Description of User Interaction SCF

The OMG IDL representation of this interface specification is contained in text files (ui_data.idl and ui_interfaces.idl contained in archive 2919805IDL.ZIP) which accompanies the present document.

Annex B (informative): Differences between this draft and 3GPP TS 29.198 R99

B.1 Interface IpUIManager

~~createEnableUINotification (appInterface-appUIManager : in IpAppUIManagerRef, eventCriteria : in TpUIEventCriteria, assignmentID : out TpAssignmentIDRef) : TpResult~~

~~createUICall (appUI : in IpAppUICallRef, uiTargetObject : in TpUITargetObject, callIdentifier : in ec::TpCallIdentifier, callLegIdentifier : in ec::TpCallLegIdentifier, userInteraction : out TpUICallIdentifierRef) : TpResult~~

~~destroyDisableUINotification (assignmentID : in TpAssignmentID) : TpResult~~

~~changeNotification (assignmentID : in TpAssignmentID, eventCriteria : in TpUIEventCriteria) : TpResult~~

~~getNotification (eventCriteria : out TpCallEventCriteriaResultSetRef) : TpResult~~

B.2 Interface IpAppUIManager

~~UserInteractionEventNotifyreportNotification (uiuserInteraction : in TpUIIdentifier, eventInfo : in TpUIEventInfo, assignmentID : in TpAssignmentID, appInterface-appUI : out IpAppUIRefRef) : TpResult~~

B.3 Interface IpUI

~~sendInfoReq (userInteractionSessionID : in TpSessionID, info : in TpUIInfo, language : in TpLanguage, variableInfo : in TpUIVariableInfoSet, repeatIndicator : in TpInt32, responseRequested : in TpUIResponseRequest, assignmentID : out TpAssignmentIDRef) : TpResult~~

~~sendInfoReq (userInteractionSessionID : in TpSessionID, info : in TpUIInfo, language : in TpLanguage, variableInfo : in TpUIVariableInfoSet, repeatIndicator : in TpInt32, responseRequested : in TpUIResponseRequest, assignmentID : out TpAssignmentIDRef) : TpResult~~

B.4 Interface IpAppUI

~~sendInfoAndCollectRes (userInteractionSessionID : in TpSessionID, assignmentID : in TpAssignmentID, response : in TpUIReport, infoCollectedInfo : in TpString) : TpResult~~

B.5 Interface IpUICall

The following method was added:

~~deleteMessageReq (userInteractionSessionID : in TpSessionID, messageID : in TpInt32, assignmentID : out TpAssignmentIDRef) : TpResult~~

B.6 Interface IpAppUICall

The following methods were added:

~~deleteMessageRes (userInteractionSessionID : in TpSessionID, response : in TpUIReport, assignmentID : in TpAssignmentID) : TpResult~~

~~deleteMessageErr (userInteractionSessionID : in TpSessionID, error : in TpUIError, assignmentID : in TpAssignmentID) : TpResult~~

B.7 All Interfaces

All methods on IpApp interfaces no longer throw exceptions.

All methods on the other interfaces throw TpCommonExceptions and individual, identified exceptions

All methods now return void or the former out parameter.

B.8 Type TpUIReport

TpUIReport

Defines the UI call-reports if a response was requested.

Name	Value	Description
P_UI_REPORT_UNDEFINED	0	Undefined report
P_UI_REPORT_ANNOUNCEMENT_ENDED P_UI_REPORT_INFO_SENT	1	Confirmation that the announcement-information <u>has ended</u> been sent
P_UI_REPORT_LEGAL_INPUT P_UI_REPORT_INFO_COLLECTED	2	Information collected., meeting the specified criteria.
P_UI_REPORT_NO_INPUT	3	No information collected. The user immediately entered the delimiter character. No valid information has been returned
P_UI_REPORT_TIMEOUT	4	No information collected. The user did not input any response before the input timeout expired
P_UI_REPORT_MESSAGE_STORED	5	A message has been stored successfully
P_UI_REPORT_MESSAGE_NOT_STORED	6	The message has not been stored successfully
P_UI_REPORT_MESSAGE_DELETED	7	A message has been deleted successfully
P_UI_REPORT_MESSAGE_NOT_DELETED	8	A message has not been deleted successfully

B.9 Type TpUIError

TpUIError

Defines the UI call-error codes.

Name	Value	Description
P_UI_ERROR_UNDEFINED	0	Undefined error
P_UI_ERROR_ILLEGAL_IDINFO	1	The specified information id(<u>InfoId, InfoData, or InfoAddress</u>) specified is invalid
P_UI_ERROR_ID_NOT_FOUND	2	A legal information id(<u>InfoId</u>) is not known to the User Interaction service
P_UI_ERROR_RESOURCE_UNAVAILABLE	3	The information resources used by the User Interaction service are unavailable, e.g. due to an overload situation.
P_UI_ERROR_ILLEGAL_RANGE	4	The values for minimum and maximum collection length are out of range
P_UI_ERROR_IMPROPER_CALLER_USER_RESPONSE	5	Improper user response
P_UI_ERROR_ABANDON	6	The specified leg is disconnected before the send information completed
P_UI_ERROR_NO_OPERATION_ACTIVE	7	There is no active User Interaction for the specified leg. Either the application did not start any User Interaction or the User Interaction was already finished when the abortAction_Req() was called.
P_UI_ERROR_NO_SPACE_AVAILABLE	8	There is no more storage capacity to record the message when the recordMessage() operation was called

Name	Value	Description
P_UI_ERROR_RESOURCE_TIMEOUT	2	The request has been accepted by the resource but it did <u>not report a result.</u>

B.10 Type TpUIEventCriteriaResult

TpUIEventCriteriaResultSetRef

Defines a reference to TpUIEventCriteriaResultSet

TPUIEventCriteriaResultSet

Defines a set of TpUIEventCriteriaResult

TPUIEventCriteriaResult

Defines a sequence of data elements that specify a requested event notification criteria with the associated assignmentID.

<u>Structure Element Name</u>	<u>Structure Element Type</u>	<u>Structure Element Description</u>
<u>EventCriteria</u>	<u>TpUIEventCriteria</u>	<u>The event criteria that were specified by the application.</u>
<u>AssignmentID</u>	<u>TpInt32</u>	<u>The associated assignmentID. This can be used to disable the notification.</u>

B.11 TpUITargetObjectType

TpUITargetObjectType

Defines the type of object where User Interaction should be performed upon.

Name	Value	Description
P_UI_TARGET_OBJECT_CALL	0	User-interaction will be performed on a complete Call.
P_UI_TARGET_OBJECT_MULTI_PARTY_CALL	1	User-interaction will be performed on a complete Multi-party Call.
P_UI_TARGET_OBJECT_CALL_LEG	2	User-interaction will be performed on a single Call Leg.

TpUITargetObject

Defines the Tagged Choice of Data Elements that specify the object to perform User Interaction on.

	Tag Element Type	
	TpUITargetObjectType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_TARGET_OBJECT_CALL	TpCallIdentifier	Call
P_UI_TARGET_OBJECT_MULTIPARTY_CALL	TpMultiPartyCallIdentifier	MultiPartyCall
P_UI_TARGET_OBJECT_CALL_LEG	TpCallLegIdentifier	CallLeg

B.12 TpUIVariableInfo

TpUIVariableInfo

Defines the Tagged Choice of Data Elements that specify the variable parts in the information to send to the user.

	Tag Element Type	
	TpUIVariablePartType	

Tag Element Value	Choice Element Type	Choice Element Name
P_UI_VARIABLE_PART_INT	TpInt32	VariablePartInteger
P_UI_VARIABLE_PART_ADDRESS	TpString	VariablePartAddress
P_UI_VARIABLE_PART_TIME	TpTime	VariablePartTime
P_UI_VARIABLE_PART_DATE	TpDate	VariablePartDate
P_UI_VARIABLE_PART_PRICE	TpPrice	VariablePartPrice

CHANGE REQUEST

⌘ **29.198-05 CR 007** ⌘ rev **-** ⌘ Current version: **4.2.0** ⌘

For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.

Proposed change affects: ⌘ (U)SIM ME/UE Radio Access Network Core Network

Title:	⌘ Correction to values of TpUInfoType		
Source:	⌘ CN5		
Work item code:	⌘ OSA1	Date:	⌘ 30/11/2001
Category:	⌘ F	Release:	⌘ REL-4
	<i>Use one of the following categories:</i> F (correction) A (corresponds to a correction in an earlier release) B (addition of feature), C (functional modification of feature) D (editorial modification) Detailed explanations of the above categories can be found in 3GPP TR 21.900.		<i>Use one of the following releases:</i> 2 (GSM Phase 2) R96 (Release 1996) R97 (Release 1997) R98 (Release 1998) R99 (Release 1999) REL-4 (Release 4) REL-5 (Release 5)

Reason for change:	⌘ TpUInfoType is an enumerated data type in the IDL. By convention, IDL enumerated types start at value 0. However, this type as documented starts with value 1.
Summary of change:	⌘ Correct TpUInfoType enumerations to start at 0.
Consequences if not approved:	⌘ Serious discrepancy between implementations based on IDL code, and implementations based on Word document, which will cause interworking problems. This change was already performed in Parlay 2.1 and has not been taken into account in the 3GPP specification set.

Clauses affected:	⌘ 11.23		
Other specs affected:	⌘ <input type="checkbox"/> Other core specifications <input type="checkbox"/> Test specifications <input type="checkbox"/> O&M Specifications	⌘	
Other comments:	⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

- 1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
- 2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under <ftp://ftp.3gpp.org/specs/> For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

- 3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

11.23 TpUIInfoType

Defines the type of the information to be send to the user.

Name	Value	Description
P_UI_INFO_ID	10	The information to be send to an end-user consists of an ID
P_UI_INFO_DATA	21	The information to be send to an end-user consists of a data string
P_UI_INFO_ADDRESS	32	The information to be send to an end-user consists of a URL.