3GPP TSG CN Plenary Meeting #17 **NP-020435**
4 - 6 September 2002, Biarritz, FRANCE

Source: **CN5 (OSA)**

Title: **Rel-5 CR 29.198-08 OSA API Part 8: Data session control**

Agenda item: **8.2**

Document for: **APPROVAL**

Doc-1st-Level	Spec	CR	Rev	Phase	Subject	Cat	Version-Current	Doc-2nd-Level	Workitem
NP-020435	29.198-08	011	-	Rel-5	Remove duplicate exception from IpDataSessionControlManager.createNotification()	F	5.0.0	N5-020633	OSA2
NP-020435	29.198-08	012	-	Rel-5	Remove P_SERVICE_INFORMATION_MISSING and P_SERVICE_FAULT_ENCOUNTERED exceptions from_DataSessionControl methods.	F	5.0.0	N5-020635	OSA2
NP-020435	29.198-08	013	-	Rel-5	Introduce new method getNotifications to correct the result type of IpDataSessionControlManager.getNotification() to permit retreival of all created notifications.	F	5.0.0	N5-020714	OSA2
NP-020435	29.198-08	014	-	Rel-5	Add P_INVALID_INTERFACE_TYPE exception to IpDataSessionControlManager.createNotification(), resulting in new createNotifications() method	F	5.0.0	N5-020715	OSA2
NP-020435	29.198-08	015	-	Rel-5	Add text to clarify requirements on support of methods	F	5.0.0	N5-020721	OSA2
NP-020435	29.198-08	016	-	Rel-5	Correction on use of NULL in Data Session Control API	A	5.0.0	N5-020764	OSA2

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020633**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **011** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Remove duplicate exception from IpDataSessionControlManager.createNotification()
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	createNotification() has exception P_INVALID_ADRESS but this error is covered by P_INVALID_CRITERIA. No other createNotification() method in any other SCF has this exception, despite our wish for similar methods in different SCFs to be aligned.
Summary of change:	⌘	Removal of P_INVALID_ADRESS exception from method signature of IpDataSessionControlManager.createNotification() This change is entirely backwards compatible according to the backwards compatibility rules agreed at previous meetings: it is permitted to remove exceptions from the SCF side, since the application does not require modification if it already has code to trap these exceptions.
Consequences if not approved:	⌘	Application developers will be forced to include code twice in their applications to trap this exception. Despite our efforts to align similar methods in the SCFs, here the code for createNotification can't be written like the same method in other SCFs.

Clauses affected:	⌘	8.4
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ 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.

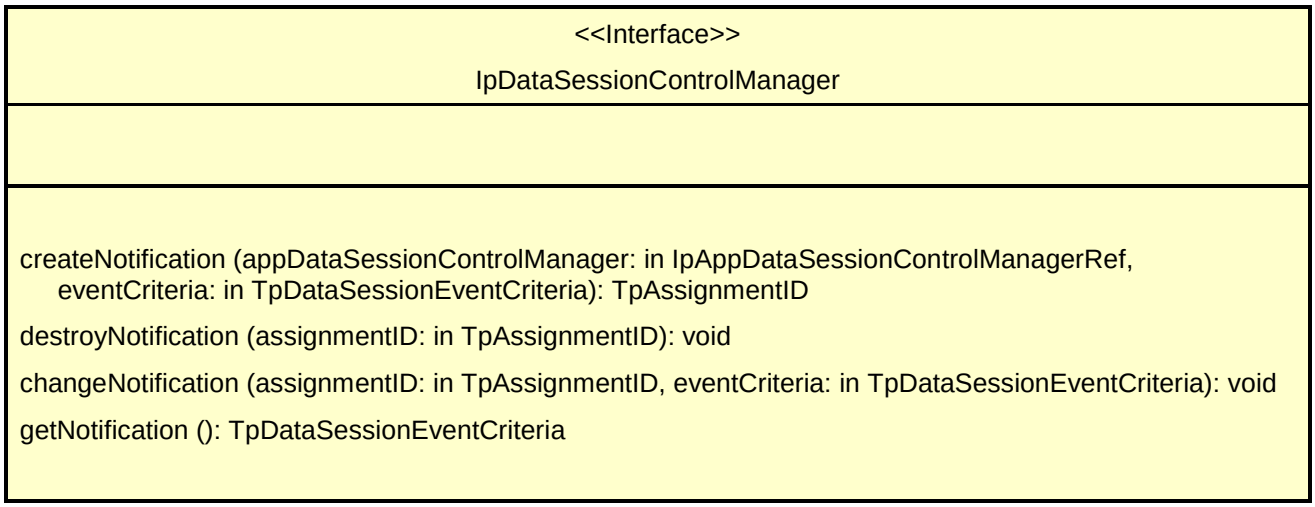
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest 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.

8.4 Interface Class IpDataSessionControlManager

Inherits from: IpService.

This interface is the "SCF manager" interface for Data Session Control.



Method

`createNotification()`

This method is used to enable data session notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of data session happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular data session it has to use the connectReq() method on the data session object. The application will get access to the data session object when it receives the reportNotification().

The createNotification method is purely intended for applications to indicate their interest to be notified when certain data session events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a data session is setup to any number starting with 800.

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.

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. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the createNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

Returns assignmentID: Specifies the ID assigned by the Data Session Manager object for this newly-enabled event notification.

Parameters

`appDataSessionControlManager: in IpAppDataSessionControlManagerRef`

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 TpDataSessionEventCriteria

Specifies the event specific criteria used by the application to define the event required. Individual addresses or address ranges may be specified for destination and/or origination. Examples of events are "Data Session set up".

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, ~~P_INVALID_ADDRESS,~~ P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

Method

destroyNotification()

This method is used by the application to disable data session notifications.

Parameters

assignmentID: in TpAssignmentID

Specifies the assignment ID given by the data session manager object when the previous createNotification() was done.

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID

Method

changeNotification()

This method is used by the application to change the event criteria introduced with the 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 TpDataSessionEventCriteria

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_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

Method

getNotification()

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

Returns eventCriteria: Specifies the event criteria used by the application to define the event required. Only events that meet these requirements are reported.

Parameters

No Parameters were identified for this method.

Returns

TpDataSessionEventCriteria

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **012** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Remove P_SERVICE_INFORMATION_MISSING and P_SERVICE_FAULT_ENCOUNTERED exceptions from DataSessionControl methods.
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	Conditions for raising P_SERVICE_INFORMATION_MISSING and P_SERVICE_FAULT_ENCOUNTERED exceptions in Data Session Control are not described in any of the method descriptions. Furthermore, these exceptions are unique to Data Session Control, when many of the methods they are applicable to are not unique. Their function could be replaced with P_TASK_REFUSED, P_RESOURCES_UNAVAILABLE or P_TASK_CANCELLED, all of which are available in TpCommonExceptions. Either these exceptions are required, in which case they should be added to each SCF, and fully described, or else they are not required, in which case they should be removed to simplify the SCF. We believe the latter is the case.
Summary of change:	⌘	Remove P_SERVICE_INFORMATION_MISSING and P_SERVICE_FAULT_ENCOUNTERED exception all methods in Data Session Control, and from the list of exceptions in clause 12. This change is entirely backwards compatible according to the backwards compatibility rules agreed at previous meetings: it is permitted to remove exceptions from the SCF side, since the application does not require modification if it already has code to trap these exceptions.
Consequences if not approved:	⌘	Application developers will be forced to include code in their applications to trap these two exceptions, which might never be used by some SCF implementations, and whose use is not fully described. Since it is not clear what conditions might lead to the raising of these exceptions, it is not clear what recovery behaviour an application should perform.

Clauses affected:	⌘	8.3, 8.4, 12

Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ 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.

8.3 Interface Class IpDataSession

Inherits from: IpService.

The Data Session interface provides basic methods for applications to control data sessions.

<<Interface>> IpDataSession
connectReq (dataSessionID : in TpSessionID, responseRequested : in TpDataSessionReportRequestSet, targetAddress : in TpAddress) : TpAssignmentID release (dataSessionID : in TpSessionID, cause : in TpDataSessionReleaseCause) : void superviseDataSessionReq (dataSessionID : in TpSessionID, treatment : in TpDataSessionSuperviseTreatment, bytes : in TpDataSessionSuperviseVolume) : void setDataSessionChargePlan (dataSessionID : in TpSessionID, dataSessionChargePlan : in TpDataSessionChargePlan) : void setAdviceOfCharge (dataSessionID : in TpSessionID, aoCInfo : in TpAoCInfo, tariffSwitch : in TpDuration) : void deassignDataSession (dataSessionID : in TpSessionID) : void continueProcessing (dataSessionID : in TpSessionID) : void

8.3.1 Method connectReq()

This asynchronous method requests the connection of a data session with the destination party (specified in the parameter TargetAddress). The Data Session object is not automatically deleted if the destination party disconnects from the data session.

Returns assignmentID : Specifies the ID assigned to the request. The same ID will be returned in the connectRes or Err. This allows the application to correlate the request and the result.

Parameters

`dataSessionID : in TpSessionID`

Specifies the session ID.

`responseRequested : in TpDataSessionReportRequestSet`

Specifies the set of observed data session events that will result in a connectRes() being generated.

`targetAddress : in TpAddress`

Specifies the address of destination party.

Returns

TpAssignmentID

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_ADDRESS, P_INVALID_SESSION_ID

8.3.2 Method release()

This method requests the release of the data session and associated objects.

Parameters

dataSessionID : in TpSessionID

Specifies the session.

cause : in TpDataSessionReleaseCause

Specifies the cause of the release.

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_SESSION_ID

8.3.3 Method superviseDataSessionReq()

The application calls this method to supervise a data session. The application can set a granted data volume for this data session. If an application calls this function before it calls a connectReq() or a user interaction function the time measurement will start as soon as the data session is connected. The Data Session object will exist after the data session has been terminated if information is required to be sent to the application at the end of the data session

Parameters

dataSessionID : in TpSessionID

Specifies the data session.

treatment : in TpDataSessionSuperviseTreatment

Specifies how the network should react after the granted data volume has been sent.

bytes : in TpDataSessionSuperviseVolume

Specifies the granted number of bytes that can be transmitted for the data session.

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_SESSION_ID

8.3.4 Method setDataSessionChargePlan()

Allows an application to include charging information in network generated CDR.

Parameters

dataSessionID : in TpSessionID

Specifies the session ID of the data session.

dataSessionChargePlan : in TpDataSessionChargePlan

Specifies the charge plan used.

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_SESSION_ID

8.3.5 Method setAdviceOfCharge()

This method allows the application to determine the charging information that will be sent to the end-users terminal.

Parameters

dataSessionID : in TpSessionID

Specifies the session ID of the data session.

aoCInfo : in TpAoCInfo

Specifies two sets of Advice of Charge parameter according to GSM.

tariffSwitch : in TpDuration

Specifies the tariff switch that signifies when the second set of AoC parameters becomes valid.

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_TIME_AND_DATE_FORMAT

8.3.6 Method deassignDataSession()

This method requests that the relationship between the application and the data session and associated objects be de-assigned. It leaves the data session in progress, however, it purges the specified data session object so that the application has no further control of data session processing. If a data session is de-assigned that has event reports, data session information reports requested, then these reports will be disabled and any related information discarded.

The application should always either release or deassign the data session when it is finished with the data session, unless dataSessionFaultDetected is received by the application.

Parameters

dataSessionID : in TpSessionID

Specifies the session ID of the data session.

Raises

TpCommonExceptions, P_INVALID_SESSION_ID

8.3.7 Method continueProcessing()

This operation continues processing of the data session. Applications can invoke this operation after session handling was interrupted due to detection of a notification or event the application subscribed its interest in.

Parameters

dataSessionID : in TpSessionID

Specifies the session ID of the data session.

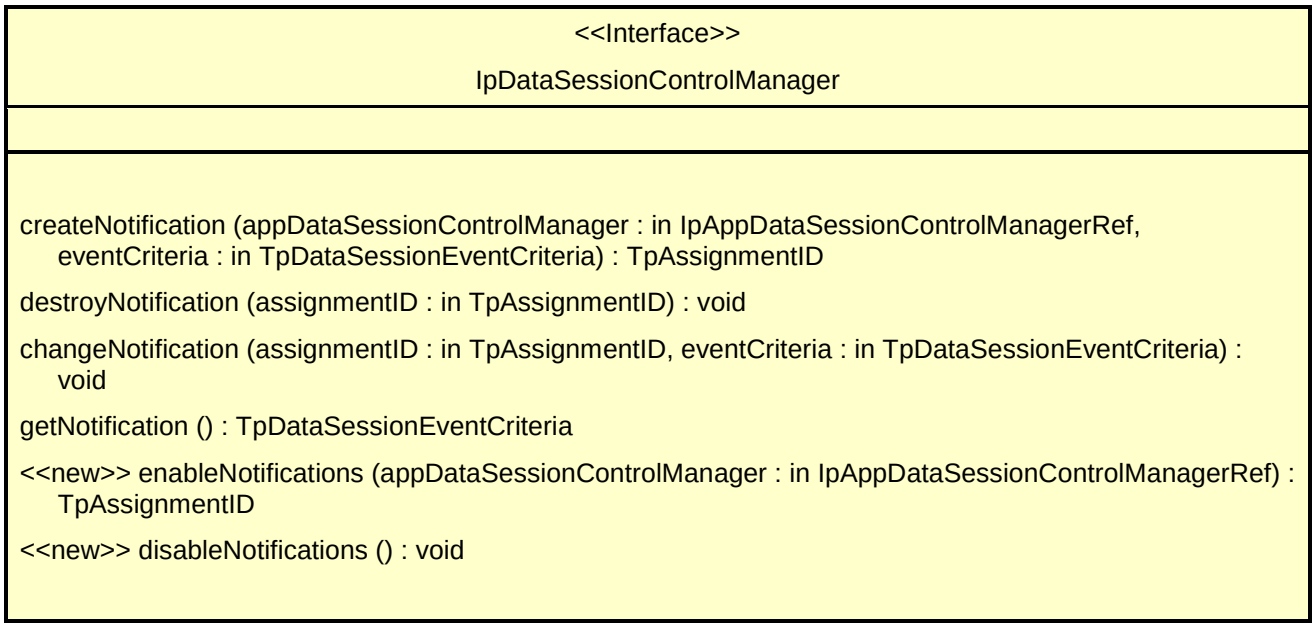
Raises

TpCommonExceptions, P_INVALID_SESSION_ID, P_INVALID_NETWORK_STATE

8.4 Interface Class IpDataSessionControlManager

Inherits from: IpService.

This interface is the 'SCF manager' interface for Data Session Control.



8.4.1 Method createNotification()

This method is used to enable data session notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of data session happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular data session it has to use the connectReq() method on the data session object. The application will get access to the data session object when it receives the reportNotification().

The createNotification method is purely intended for applications to indicate their interest to be notified when certain data session events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a data session is setup to any number starting with 800.

If some application already requested notifications with criteria that overlap the specified criteria or the specified criteria overlap with criteria already present in the network (when provisioned from within the network), 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.

If a notification is requested by an application with monitor mode set to notify, then there is no need to check the rest of the criteria for overlapping with any existing request as the notify mode does not give control of a data session. Only one application can place an interrupt request if the criteria overlaps.

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. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the createNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

Returns assignmentID : Specifies the ID assigned by the Data Session Manager object for this newly-enabled event notification.

Parameters

`appDataSessionControlManager : in IpAppDataSessionControlManagerRef`

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 TpDataSessionEventCriteria`

Specifies the event specific criteria used by the application to define the event required. Individual addresses or address ranges may be specified for destination and/or origination. Examples of events are "Data Session set up".

Returns

TpAssignmentID

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_ADDRESS, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

8.4.2 Method destroyNotification()

This method is used by the application to disable data session notifications. This method only applies to notifications created with createNotification().

Parameters

assignmentID : in TpAssignmentID

Specifies the assignment ID given by the data session manager object when the previous createNotification() was done.

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID

8.4.3 Method changeNotification()

This method is used by the application to change the event criteria introduced with the 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 TpDataSessionEventCriteria

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_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

8.4.4 Method getNotification()

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

Returns eventCriteria : Specifies the event criteria used by the application to define the event required. Only events that meet these requirements are reported.

Parameters

No Parameters were identified for this method

Returns

TpDataSessionEventCriteria

Raises

TpCommonExceptions, ~~P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED,~~ P_INVALID_NETWORK_STATE

8.4.5 Method <<new>> enableNotifications()

This method is used to indicate that the application is able to receive which are provisioned from within the network (i.e. these notifications are NOT set using createNotification() but via, for instance, a network management system). If notifications provisioned for this application are created or changed, the application is unaware of this until the notification is reported.

If the same application requests to enable notifications for a second time with a different IpAppDataSessionControlManager reference (i.e. without first disabling them), the second callback will be treated as an additional callback. This means that the callback wil only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).

When this method is used, it is still possible to use createNotification() for service provider provisioned notifications on the same interface as long as the criteria in the network and provided by createNotification() do not overlap. However, it is NOT recommended to use both mechanisms on the same service manager.

The methods changeNotification(), getNotification(), and destroyNotification() do not apply to notifications provisoned in the network and enabled using enableNotifications(). These only apply to notifications created using createNotification().

Returns assignmentID: Specifies the ID assigned by the manager interface for this operation. This ID is contained in any reportNotification() that relates to notifications provisioned from within the network. Repeated calls to enableNotifications() return the same assignment ID.

Parameters

appDataSessionControlManager : in IpAppDataSessionControlManagerRef

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.

Returns

TpAssignmentID

Raises

TpCommonExceptions

8.4.6 Method <<new>> disableNotifications()

This method is used to indicate that the application is not able to receive notifications for which the provisioning has been done from within the network. (i.e. these notifications that are NOT set using createNotification() but via, for instance, a network management system). After this method is called, no such notifications are reported anymore.

Parameters

No Parameters were identified for this method

Raises

TpCommonExceptions

12Exception Classes

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

Name	Description
P_SERVICE_INFORMATION_MISSING	Information relating to the Data Session Control SCF could not be found
P_SERVICE_FAULT_ENCOUNTERED	Fault detected in the Data Session Control SCF

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

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020714**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **013** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Introduce new method getNotifications to correct the result type of IpDataSessionControlManager.getNotification() to permit retreival of all created notifications.
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	In description of IpDataSessionControlManager. getNotification () the result is given as a TpDataSessionEventCriteria data type, which permits only one created notification to be returned. There is no means to select which one, nor to return the entire list (more than one non-overlapping notification can be created). The other getNotification() methods in the other SCFs all return a list of notification criteria, sorted by AssignmentID. This one should be no different. In order to make this change in a backwards compatible manner, getNotification() is deprecated and a new method getNotifications is introduced. This breaks with existing naming convention.
Summary of change:	⌘	Deprecate IpIpDataSessionControlManager.getNotification(). Introduce new getNotifications() to change result type from TpDataSessionEventCriteria to TpDataSessionEventCriteriaResultSet and add this data type definition.
Consequences if not approved:	⌘	This method can't be used as described ! There is no means to return the entire list of requested notifications, nor to know which notification is being returned: the first created one, the most recent created one, cycling through the list etc.

Clauses affected:	⌘	8.4, 11
Other specs Affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ 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.

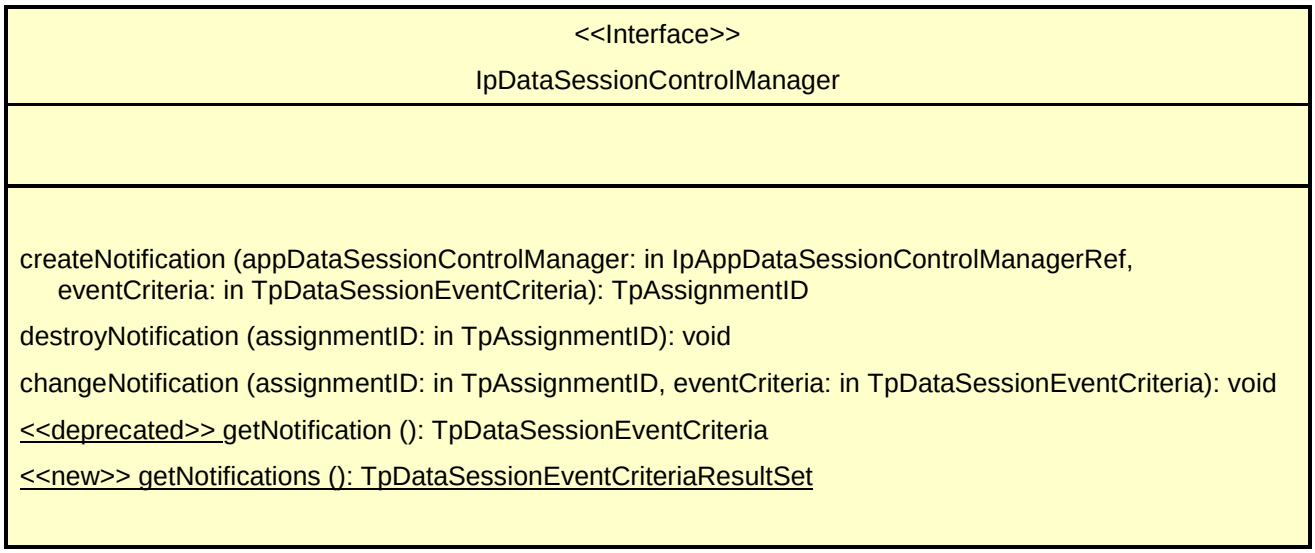
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest 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.

8.4 Interface Class IpDataSessionControlManager

Inherits from: IpService.

This interface is the "SCF manager" interface for Data Session Control.



Method

`createNotification()`

This method is used to enable data session notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of data session happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular data session it has to use the connectReq() method on the data session object. The application will get access to the data session object when it receives the reportNotification().

The createNotification method is purely intended for applications to indicate their interest to be notified when certain data session events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a data session is setup to any number starting with 800.

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.

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. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the createNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

Returns assignmentID: Specifies the ID assigned by the Data Session Manager object for this newly-enabled event notification.

Parameters

`appDataSessionControlManager: in IpAppDataSessionControlManagerRef`

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 TpDataSessionEventCriteria

Specifies the event specific criteria used by the application to define the event required. Individual addresses or address ranges may be specified for destination and/or origination. Examples of events are "Data Session set up".

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ADDRESS, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

Method

destroyNotification()

This method is used by the application to disable data session notifications.

Parameters

assignmentID: in TpAssignmentID

Specifies the assignment ID given by the data session manager object when the previous createNotification() was done.

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID

Method

changeNotification()

This method is used by the application to change the event criteria introduced with the 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 TpDataSessionEventCriteria

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_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

Method

<<deprecated>> getNotification()

This method is deprecated and its use is discouraged. It will be removed in a later release. It is replaced with getNotifications.

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

Returns eventCriteria: Specifies the event criteria used by the application to define the event required. Only events that meet these requirements are reported.

Parameters

No Parameters were identified for this method.

Returns

TpDataSessionEventCriteria

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE

Method

<<new>> getNotifications()

This method replaces getNotification()

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

Returns eventCriteria: the list of event criteria for the notifications requested by the application. If there is no information to return (e.g. no notifications requested by the application), an empty set (zero length) is returned.

Parameters

No Parameters were identified for this method.

Returns

TpDataSessionEventCriteriaResultSet

Raises

TpCommonExceptions, P_INVALID_NETWORK_STATE

11 Data Definitions

All data types referenced but not defined in this clause are common data definitions which may be found in ES 201 915-2.

11.1 Data Session Control Data Definitions

11.1.1 IpAppDataSession

Defines the address of an `IpAppDataSession` Interface.

11.1.2 IpAppDataSessionRef

Defines a Reference to type `IpAppDataSession`

11.1.3 IpAppDataSessionControlManager

Defines the address of an `IpAppDataSessionControlManager` Interface.

11.1.4 IpAppDataSessionControlManagerRef

Defines a Reference to type `IpAppDataSessionControlManager`.

11.1.5 IpDataSession

Defines the address of an `IpDataSession` Interface.

11.1.6 IpDataSessionRef

Defines a Reference to type `IpDataSession`.

11.1.7 IpDataSessionControlManager

Defines the address of an `IpDataSessionControlManager` Interface.

11.1.8 IpDataSessionControlManagerRef

Defines a Reference to type `IpDataSessionControlManager`.

11.2 Event Notification data definitions

11.2.1 TpDataSessionEventName

Defines the names of events being notified with a new call request. The following events are supported. The values may be combined by a logical "OR" function when requesting the notifications. Additional events that can be requested/received during the call process are found in the TpDataSessionReportType data-type.

Name	Value	Description
P_EVENT_NAME_UNDEFINED	0	Undefined
P_EVENT_DSCS_SETUP	1	The data session is going to be setup.
P_EVENT_DSCS_ESTABLISHED	2	The data session is established by the network.
P_EVENT_DSCS_QOS_CHANGED	4	A change in QoS class has taken place during the life of the data session.

11.2.2 TpDataSessionMonitorMode

Defines the mode that the call will monitor for events, or the mode that the call is in following a detected event.

Name	Value	Description
P_DATA_SESSION_MONITOR_MODE_INTERRUPT	0	The data session event is intercepted by the data session control service and data session establishment is interrupted. The application is notified of the event and data session establishment resumes following an appropriate API call or network event (such as a data session release)
P_DATA_SESSION_MONITOR_MODE_NOTIFY	1	The data session event is detected by the data session control service but not intercepted. The application is notified of the event and data session establishment continues
P_DATA_SESSION_MONITOR_MODE_DO_NOT_MONITOR	2	Do not monitor for the event

11.2.3 TpDataSessionEventCriteria

Defines the Sequence of Data Elements that specify the criteria for an event notification.

Of the addresses only the Plan and the AddrString are used for the purpose of matching the notifications against the criteria.

Sequence Element Name	Sequence Element Type	Description
DestinationAddress	TpAddressRange	Defines the destination address or address range for which the notification is requested.
OriginatingAddress	TpAddressRange	Defines the origination address or an address range for which the notification is requested.
DataSessionEventName	TpDataSessionEventName	Name of the event(s)
MonitorMode	TpDataSessionMonitorMode	Defines the mode that the Data Session is in following the notification. Monitor mode P_DATA_SESSION_MONITOR_MODE_DO_NOT_MONITOR is not a legal value here.

11.2.4 TpDataSessionEventInfo

Defines the Sequence of Data Elements that specify the information returned to the application in a Data Session event notification.

Sequence Element Name	Sequence Element Type	Description
DestinationAddress	TpAddress	Defines the destination address for which the notification is reported.
OriginatingAddress	TpAddress	Defines the origination address for which the notification is reported.
DataSessionEventName	TpDataSessionEventName	Name of the event(s)
MonitorMode	TpDataSessionMonitorMode	Defines the mode in which the Data Session is reporting the notification. Monitor mode P_DATA_SESSION_MONITOR_MODE_DO_NOT_MONITOR is not a legal value here.
QoSClass	TpDataSessionQosClass	Defines the Quality of Service (QoS) class for the Data Session. QoSClass NULL is not a legal value when DataSessionEventName is set to P_EVENT_DSCS_QOS_CHANGED. For this particular event, the QoSClass defines the new QoS class effective after the change.

11.2.5 TpDataSessionQosClass

Defines the Quality of Service (QoS) classes for a data session.

Name	Value	Description
P_DATA_SESSION_QOS_CLASS_CONVERSATIONAL	0	Specifies the Conversational QoS class, as specified in TS 123 107.
P_DATA_SESSION_QOS_CLASS_STREAMING	1	Specifies the Streaming QoS class, as specified in TS 123 107.
P_DATA_SESSION_QOS_CLASS_INTERACTIVE	2	Specifies the Interactive QoS class, as specified in TS 123 107.
P_DATA_SESSION_QOS_CLASS_BACKGROUND	3	Specifies the Background QoS class, as specified in TS 123 107.

11.2.6 TpDataSessionChargePlan

Defines the Sequence of Data Elements that specify the charge plan for the call.

Sequence Element Name	Sequence Element Type	Description
ChargeOrderType	TpDataSessionChargeOrder	Charge order
Currency	TpString	Currency unit according to ISO-4217:1995
AdditionalInfo	TpString	Descriptive string which is sent to the billing system without prior evaluation. Could be included in the ticket.

Valid Currencies are:

ADP, AED, AFA, ALL, AMD, ANG, AON, AOR, ARS, ATS, AUD, AWG, AZM, BAM,

BBD, BDT, BEF, BGL, BGN, BHD, BIF, BMD, BND, BOB, BOV, BRL, BSD, BTN,

BWP, BYB, BZD, CAD, CDF, CHF, CLF, CLP, CNY, COP, CRC, CUP, CVE, CYP,

CZK, DEM, DJF, DKK, DOP, DZD, ECS, ECV, EEK, EGP, ERN, ESP, ETB, EUR,

FIM, FJD, FKP, FRF, GBP, GEL, GHC, GIP, GMD, GNF, GRD, GTQ, GWP, GYD,

HKD, HNL, HRK, HTG, HUF, IDR, IEP, ILS, INR, IQD, IRR, ISK, ITL, JMD,

JOD, JPY, KES, KGS, KHR, KMF, KPW, KRW, KWD, KYD, KZT, LAK, LBP, LKR,

LRD, LSL, LTL, LUF, LVL, LYD, MAD, MDL, MGF, MKD, MMK, MNT, MOP, MRO,

MTL, MUR, MVR, MWK, MXN, MXV, MYR, MZM, NAD, NGN, NIO, NLG, NOK, NPR,

NZD, OMR, PAB, PEN, PGK, PHP, PKR, PLN, PTE, PYG, QAR, ROL, RUB, RUR,

RWF, SAR, SBD, SCR, SDD, SEK, SGD, SHP, SIT, SKK, SLL, SOS, SRG, STD,

SVC, SYP, SZL, THB, TJR, TMM, TND, TOP, TPE, TRL, TTD, TWD, TZS, UAH,

UGX, USD, USN, USS, UYU, UZS, VEB, VND, VUV, WST, XAF, XAG, XAU, XBA,

XBB, XBC, XBD, XCD, XDR, XFO, XFU, XOF, XPD, XPF, XPT, XTS, XXX, YER,

YUM, ZAL, ZAR, ZMK, ZRN, ZWD.

XXX is used for transactions where no currency is involved.

11.2.7 TpDataSessionChargeOrder

Defines the Tagged Choice of Data Elements that specify the charge plan for the call.

	Tag Element Type	
	TpDataSessionChargeOrderCategory	

Tag Element Value	Choice Element Type	Choice Element Name
P_DATA_SESSION_CHARGE_PER_VOLUME	TpChargePerVolume	ChargePerVolume
P_DATA_SESSION_CHARGE_NETWORK	TpString	NetworkCharge

11.2.8 TpDataSessionChargeOrderCategory

Name	Value	Description
P_DATA_SESSION_CHARGE_PER_VOLUME	0	Charge per volume
P_DATA_SESSION_CHARGE_NETWORK	1	Operator specific charge plan specification, e.g. charging table name/charging table entry

11.2.9 TpChargePerVolume

Defines the Sequence of Data Elements that specify the time based charging information. The volume is the sum of uplink and downlink transfer data volumes.

Sequence Element Name	Sequence Element Type	Description
InitialCharge	TpInt32	Initial charge amount (in currency units * 0.0001)
CurrentChargePerKilobyte	TpInt32	Current tariff (in currency units * 0.0001)
NextChargePerKilobyte	TpInt32	Next tariff (in currency units * 0.0001) after tariff switch. Only used in setAdviceOfCharge()

11.2.10 TpDataSessionIdentifier

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

Sequence Element Name	Sequence Element Type	Sequence Element Description
DataSessionReference	IpDataSessionRef	This element specifies the interface reference for the Data Session object.
DataSessionID	TpSessionID	This element specifies the data session ID of the Data Session.

11.2.11 TpDataSessionError

Defines the Sequence of Data Elements that specify the additional information relating to a call error.

Sequence Element Name	Sequence Element Type
ErrorTime	TpDateAndTime
ErrorType	TpDataSessionErrorType
AdditionalErrorInfo	TpDataSessionAdditionalErrorInfo

11.2.12 TpDataSessionAdditionalErrorInfo

Defines the Tagged Choice of Data Elements that specify additional Data Session error and Data Session error specific information.

	Tag Element Type	
	TpDataSessionErrorType	

Tag Element Value	Choice Element Type	Choice Element Name
P_DATA_SESSION_ERROR_UNDEFINED	NULL	Undefined
P_DATA_SESSION_ERROR_INVALID_ADDRESS	TpAddressError	DataSessionErrorInvalidAddress
P_DATA_SESSION_ERROR_INVALID_STATE	NULL	Undefined

11.2.13 TpDataSessionErrorType

Defines a specific Data Session error.

Name	Value	Description
P_DATA_SESSION_ERROR_UNDEFINED	0	Undefined; the method failed or was refused, but no specific reason can be given.
P_DATA_SESSION_ERROR_INVALID_ADDRESS	1	The operation failed because an invalid address was given
P_DATA_SESSION_ERROR_INVALID_STATE	2	The data session was not in a valid state for the requested operation

11.2.14 TpDataSessionFault

Defines the cause of the data session fault detected.

Name	Value	Description
P_DATA_SESSION_FAULT_UNDEFINED	0	Undefined
P_DATA_SESSION_USER_ABORTED	1	User has finalised the data session before any message could be sent by the application
P_DATA_SESSION_TIMEOUT_ON_RELEASE	2	This fault occurs when the final report has been sent to the application, but the application did not explicitly release data session object, within a specified time. The timer value is operator specific.
P_DATA_SESSION_TIMEOUT_ON_INTERRUPT	3	This fault occurs when the application did not instruct the gateway how to handle the call within a specified time, after the gateway reported an event that was requested by the application in interrupt mode. The timer value is operator specific.

11.2.15 TpDataSessionReleaseCause

Defines the Sequence of Data Elements that specify the cause of the release of a data session.

Sequence Element Name	Sequence Element Type
Value	TpInt32
Location	TpInt32
NOTE: the Value and Location are specified as in ITU-T Recommendation Q.850.	

11.2.16 TpDataSessionSuperviseVolume

Defines the Sequence of Data Elements that specify the amount of volume that is allowed to be transmitted for the specific connection.

Sequence Element Name	Sequence Element Type	Sequence Element Description
VolumeQuantity	TpInt32	This data type is identical to a TpInt32, and defines the quantity of the granted volume that can be transmitted for the specific connection. The volume specifies the sum of uplink and downlink transfer data volumes.
VolumeUnit	TpInt32	In Order to enlarge the range of the volume quantity value the exponent of a scaling factor (10^VolumeUnit) is provided. When the unit is for example in kilobytes, VolumeUnit shall be set to 3.

11.2.17 TpDataSessionSuperviseReport

Defines the responses from the data session control service for calls that are supervised. The values may be combined by a logical "OR" function.

Name	Value	Description
P_DATA_SESSION_SUPERVISE_VOLUME_REACHED	01h	The maximum volume has been reached.
P_DATA_SESSION_SUPERVISE_DATA_SESSION_ENDED	02h	The data session has ended, either due to data session party to reach of maximum volume or calling or called release.
P_DATA_SESSION_SUPERVISE_MESSAGE_SENT	04h	A warning message has been sent.

11.2.18 TpDataSessionSuperviseTreatment

Defines the treatment of the call by the data session control service when the supervised volume is reached. The values may be combined by a logical "OR" function.

Name	Value	Description
P_DATA_SESSION_SUPERVISE_RELEASE	01h	Release the data session when the data session supervision volume is reached.
P_DATA_SESSION_SUPERVISE_RESPOND	02h	Notify the application when the call supervision volume is reached.
P_DATA_SESSION_SUPERVISE_INFORM	04h	Send a warning message to the originating party when the maximum volume is reached. If data session release is requested, then the data session will be released following the message after an administered time period

11.2.19 TpDataSessionReport

Defines the Sequence of Data Elements that specify the data session report specific information.

Sequence Element Name	Sequence Element Type
MonitorMode	TpDataSessionMonitorMode
DataSessionEventTime	TpDateAndTime
DataSessionReportType	TpDataSessionReportType
AdditionalReportInfo	TpDataSessionAdditionalReportInfo

11.2.20 TpDataSessionAdditionalReportInfo

Defines the Tagged Choice of Data Elements that specify additional data session report information for certain types of reports.

	Tag Element Type	
	TpDataSessionReportType	

Tag Element Value	Choice Element Type	Choice Element Name
P_DATA_SESSION_REPORT_UNDEFINED	NULL	Undefined
P_DATA_SESSION_REPORT_CONNECTED	NULL	Undefined
P_DATA_SESSION_REPORT_DISCONNECT	TpDataSessionReleaseCause	DataSessionDisconnect

11.2.21 TpDataSessionReportRequest

Defines the Sequence of Data Elements that specify the criteria relating to data session report requests.

Sequence Element Name	Sequence Element Type
MonitorMode	TpDataSessionMonitorMode
DataSessionReportType	TpDataSessionReportType

11.2.22 TpDataSessionReportRequestSet

Defines a Numbered Set of Data Elements of TpDataSessionReportRequest.

11.2.23 TpDataSessionReportType

Defines a specific data session event report type.

Name	Value	Description
P_DATA_SESSION_REPORT_UNDEFINED	0	Undefined
P_DATA_SESSION_REPORT_CONNECTED	1	Data session established.
P_DATA_SESSION_REPORT_DISCONNECT	2	Data session disconnect requested by data session party

11.2.24 TpDataSessionEventCriteriaResult

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

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

11.2.25 TpDataSessionEventCriteriaResultSet

Defines a set of TpDataSessionEventCriteriaResult.

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **014** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Add P_INVALID_INTERFACE_TYPE exception to IpDataSessionControlManager.createNotification(), resulting in new createNotifications() method
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	createNotification() is missing exception P_INVALID_INTERFACE_TYPE. In all methods, which used one or more interface data type in parameter list, P_INVALID_INTERFACE_TYPE is used. Without this exception, there is no means for the SCF to report a meaningful error to the application if it has packed an invalid interface reference into this parameter. Adding of P_INVALID_INTERFACE_TYPE exception to method signature of IpDataSessionControlManager.createNotification() would result in a non-backwards compatible change. Therefore the solution is to deprecate the existing createNotification() method, and replace it with an almost identical createNotifications() method, with this exception added, and with certain other exceptions, already agreed as not necessary, removed.
Summary of change:	⌘	Deprecate IpDataSessionControlManager.createNotification(). Add new method IpDataSessionControlManager.createNotifications(), identical to existing createNotification(), but with exceptions P_INVALID_INTERFACE_TYPE, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED and P_INVALID_ADDRESS exceptions removed.
Consequences if not approved:	⌘	Application developers won't be able to catch exactly this exception. Despite our efforts to align similar methods in the SCFs, here the code for createNotification can't be written like the same method in other SCFs.

Clauses affected:	⌘	8.4
Other specs affected:	⌘	☐ Other core specifications ☐ Test specifications ☐ 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.

8.4 Interface Class IpDataSessionControlManager

Inherits from: IpService.

This interface is the 'SCF manager' interface for Data Session Control.

<<Interface>> IpDataSessionControlManager
<<deprecated>> createNotification (appDataSessionControlManager : in IpAppDataSessionControlManagerRef, eventCriteria : in TpDataSessionEventCriteria) : TpAssignmentID destroyNotification (assignmentID : in TpAssignmentID) : void changeNotification (assignmentID : in TpAssignmentID, eventCriteria : in TpDataSessionEventCriteria) : void getNotification () : TpDataSessionEventCriteria <<new>> enableNotifications (appDataSessionControlManager : in IpAppDataSessionControlManagerRef) : TpAssignmentID <<new>> disableNotifications () : void <<new>> createNotifications (appDataSessionControlManager : in IpAppDataSessionControlManagerRef, eventCriteria : in TpDataSessionEventCriteria) : TpAssignmentID

8.4.1 Method <<deprecated>> createNotification()

This method is deprecated and will be removed in a later release. It is replaced with createNotifications().

This method is used to enable data session notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of data session happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular data session it has to use the connectReq() method on the data session object. The application will get access to the data session object when it receives the reportNotification().

The createNotification method is purely intended for applications to indicate their interest to be notified when certain data session events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a data session is setup to any number starting with 800.

If some application already requested notifications with criteria that overlap the specified criteria or the specified criteria overlap with criteria already present in the network (when provisioned from within the network), 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.

If a notification is requested by an application with monitor mode set to notify, then there is no need to check the rest of the criteria for overlapping with any existing request as the notify mode does not give control of a data session. Only one application can place an interrupt request if the criteria overlaps.

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. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the createNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

Returns assignmentID : Specifies the ID assigned by the Data Session Manager object for this newly-enabled event notification.

Parameters

appDataSessionControlManager : in IpAppDataSessionControlManagerRef

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 TpDataSessionEventCriteria

Specifies the event specific criteria used by the application to define the event required. Individual addresses or address ranges may be specified for destination and/or origination. Examples of events are "Data Session set up".

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ADDRESS, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

8.4.2 Method destroyNotification()

This method is used by the application to disable data session notifications. This method only applies to notifications created with createNotification().

Parameters

assignmentID : in TpAssignmentID

Specifies the assignment ID given by the data session manager object when the previous createNotification() was done.

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID

8.4.3 Method changeNotification()

This method is used by the application to change the event criteria introduced with the 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 TpDataSessionEventCriteria

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_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE, P_INVALID_ASSIGNMENT_ID, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE

8.4.4 Method getNotification()

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

Returns eventCriteria : Specifies the event criteria used by the application to define the event required. Only events that meet these requirements are reported.

Parameters

No Parameters were identified for this method

Returns

TpDataSessionEventCriteria

Raises

TpCommonExceptions, P_SERVICE_INFORMATION_MISSING, P_SERVICE_FAULT_ENCOUNTERED, P_INVALID_NETWORK_STATE

8.4.5 Method <<new>> enableNotifications()

This method is used to indicate that the application is able to receive which are provisioned from within the network (i.e. these notifications are NOT set using createNotification() but via, for instance, a network management system). If notifications provisioned for this application are created or changed, the application is unaware of this until the notification is reported.

If the same application requests to enable notifications for a second time with a different IpAppDataSessionControlManager reference (i.e. without first disabling them), the second callback will be treated as an additional callback. This means that the callback wil only be used in cases when the first callback specified by the application is unable to handle the callEventNotify (e.g. due to overload or failure).

When this method is used, it is still possible to use createNotification() for service provider provisioned notifications on the same interface as long as the criteria in the network and provided by createNotification() do not overlap. However, it is NOT recommended to use both mechanisms on the same service manager.

The methods changeNotification(), getNotification(), and destroyNotification() do not apply to notifications provisoned in the network and enabled using enableNotifications(). These only apply to notifications created using createNotification().

Returns assignmentID: Specifies the ID assigned by the manager interface for this operation. This ID is contained in any reportNotification() that relates to notifications provisioned from within the network. Repeated calls to enableNotifications() return the same assignment ID.

Parameters

appDataSessionControlManager : in IpAppDataSessionControlManagerRef

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.

Returns

TpAssignmentID

Raises

TpCommonExceptions

8.4.6 Method <<new>> disableNotifications()

This method is used to indicate that the application is not able to receive notifications for which the provisioning has been done from within the network. (i.e. these notifications that are NOT set using createNotification() but via, for instance, a network management system). After this method is called, no such notifications are reported anymore.

Parameters

No Parameters were identified for this method

Raises

TpCommonExceptions

8.4.7 Method <<new>> createNotifications()

This method is deprecated and will be removed in a later release. It is replaced with createNotifications().

This method is used to enable data session notifications so that events can be sent to the application. This is the first step an application has to do to get initial notifications of data session happening in the network. When such an event happens, the application will be informed by reportNotification(). In case the application is interested in other events during the context of a particular data session it has to use the connectReq() method on the data session object. The application will get access to the data session object when it receives the reportNotification().

The createNotification method is purely intended for applications to indicate their interest to be notified when certain data session events take place. It is possible to subscribe to a certain event for a whole range of addresses, e.g. the application can indicate it wishes to be informed when a data session is setup to any number starting with 800.

If some application already requested notifications with criteria that overlap the specified criteria or the specified criteria overlap with criteria already present in the network (when provisioned from within the network), 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.

If a notification is requested by an application with monitor mode set to notify, then there is no need to check the rest of the criteria for overlapping with any existing request as the notify mode does not give control of a data session. Only one application can place an interrupt request if the criteria overlaps.

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. Both notifications will share the same assignmentID. The gateway will always use the most recent callback. In case this most recent callback fails the second most recent is used. In case the createNotification contains no callback, at the moment the application needs to be informed the gateway will use as callback the callback that has been registered by setCallback().

Returns assignmentID : Specifies the ID assigned by the Data Session Manager object for this newly-enabled event notification.

Parameters

appDataSessionControlManager : in IpAppDataSessionControlManagerRef

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 TpDataSessionEventCriteria

Specifies the event specific criteria used by the application to define the event required. Individual addresses or address ranges may be specified for destination and/or origination. Examples of events are "Data Session set up".

Returns

TpAssignmentID

Raises

TpCommonExceptions, P_INVALID_NETWORK_STATE, P_INVALID_CRITERIA, P_INVALID_EVENT_TYPE, P_INVALID_INTERFACE_TYPE

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020721**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **015** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Add text to clarify requirements on support of methods
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**F**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	It is not clear in the OSA Specifications what exactly is meant by support of a method: is it sufficient to include such code as to respond correctly to a method invocation with the exception P_METHOD_NOT_SUPPORTED, or is it required to support the functionality described and defined by the method?
Summary of change:	⌘	Add text to clause 4 to indicate that support or implementation of a method requires that the functionality of the method be supported or implemented.
Consequences if not approved:	⌘	Different vendors and application developers will each build equipment and applications which they claim to be conformant, but which will never interwork.

Clauses affected:	⌘	4
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ 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.

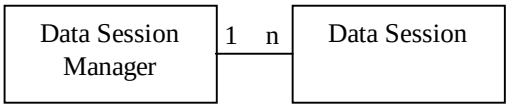
4 Data Session Control SCF

The Data Session Control network SCF consists of two interfaces:

1) Data Session manager, containing management functions for data session related issues;

2) Data Session, containing methods to control a session.

A session can be controlled by one Data Session Manager only. Data Session Manager can control several sessions.



NOTE: The term "data session" is used in a broad sense to describe a data connection/session. For example, it comprises a PDP context in GPRS.

Figure 1: Data Session control interfaces usage relationship

The Data Session Control SCFs are described in terms of the methods in the Data Session Control interfaces. Table 1 gives an overview of the Data Session Control methods and to which interfaces these methods belong.

Table 1: Overview of Data Session Control interfaces and their methods

Data Session Manager	Data Session
createNotification	connectReq
destroyNotification	connectRes
dataSessionNotificationInterrupted	connectErr
dataSessionNotificationContinued	release
reportNotification	superviseDataSessionReq
dataSessionAborted	superviseDataSessionRes
getNotification	superviseDataSessionErr
changeNotification	dataSessionFaultDetected
enableNotifications	setAdviceofCharge
disableNotifications	setDataSessionChargePlan

The session manager interface provides the management functions to the data session service capability features. The application programmer can use this interface to enable or disable data session-related event notifications.

The following clauses describe each aspect of the Data Session Control Service Capability Feature (SCF).

The order is as follows:

- the Sequence diagrams give the reader a practical idea of how each of the SCF is implemented;
- the Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another;
- the Interface specification clause describes in detail each of the interfaces shown within the Class diagram part;
- the State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions;
- the Data definitions section show a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part of this specification.

An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method. Where a method is not supported by an implementation of a Service interface, the exception P_METHOD_NOT_SUPPORTED shall be returned to any call of that method.

joint API group (Parlay, ETSI Project OSA, 3GPP TSG_CN WG5) **N5-020764**
Meeting #19, Montreal, CANADA, 8 – 12 July 2002

CR-Form-v5

CHANGE REQUEST

⌘ **29.198-08** **CR** **016** ⌘ **rev** **-** ⌘ Current version: **5.0.0** ⌘

*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 **X**

Title:	⌘	Correction on use of NULL in Data Session Control API
Source:	⌘	CN5
Work item code:	⌘	OSA2
Date:	⌘	12/07/2002
Category:	⌘	**A**
Release:	⌘	*REL-5*

Use one of the following categories:
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.

Use one of the following releases:
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:	⌘	OMG IDL does not support NULL as a valid value for a data type; attempts to send a null value result in a marshalling exception and a gateway can never receive the call.
Summary of change:	⌘	Use of null for dataSessionReference parameter in reportNotification method modified to define appropriate behaviour in NOTIFY mode
Consequences if not approved:	⌘	Failure to correct the API shall result in vendor specific interpretation and interoperability issues.

Clauses affected:	⌘	8.2;
Other specs affected:	⌘	☐ Other core specifications ⌘ ☐ Test specifications ☐ O&M Specifications
Other comments:	⌘	Mirror of Rel-4 CR 29.198-08 in N5-020761.

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:

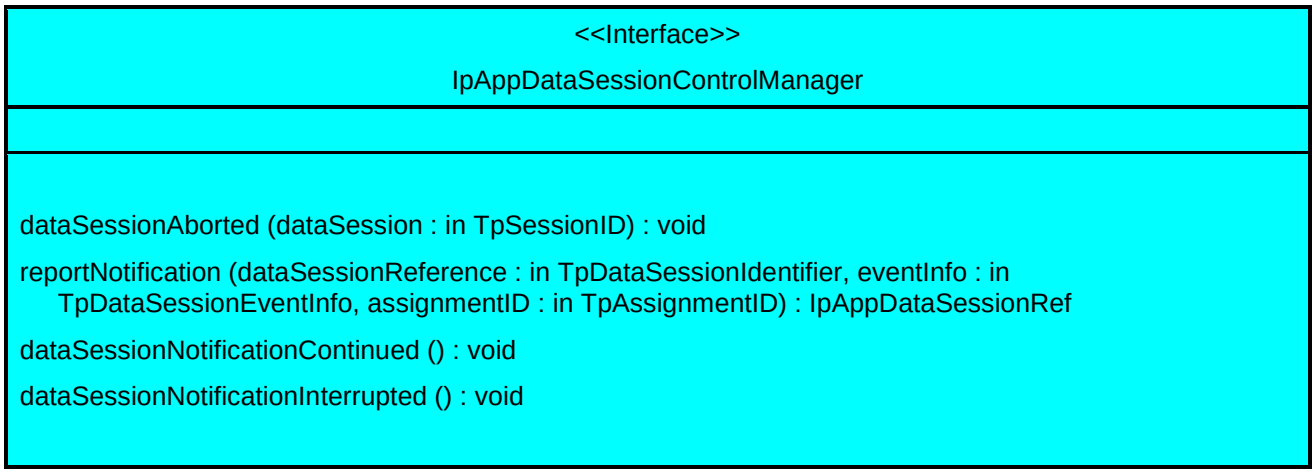
1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.
2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes")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

********** START OF FIRST CHANGE **********

8.2 Interface Class IpAppDataSessionControlManager

Inherits from: IpInterface.

The data session control manager application interface provides the application data session control management functions to the data session control SCF.



Method

dataSessionAborted()

This method indicates to the application that the Data Session object has aborted or terminated abnormally. No further communication will be possible between the Data Session object and the application.

Parameters

dataSession : in TpSessionID

Specifies the session ID of the data session that has aborted or terminated abnormally.

Method

reportNotification()

This method notifies the application of the arrival of a data session-related event.

Returns appDataSession : Specifies a reference to the application object which implements the callback interface for the new data session. If the application has previously explicitly passed a reference to the IpAppDataSession interface using a setCallback() invocation, this parameter may be null, or if supplied must be the same as that provided during the setCallback().

This parameter will be null if the notification is in NOTIFY mode.

Parameters

dataSessionReference : in TpDataSessionIdentifier

Specifies the session ID and the reference to the Data Session object to which the notification relates. ~~This parameter will be null i~~If the notification is being given in NOTIFY mode, this parameter shall be ignored by the application client

implementation, and consequently the implementation of the SCS entity invoking reportNotification may populate this parameter as it chooses..

eventInfo : in TpDataSessionEventInfo

Specifies data associated with this event. This data includes the destination address provided by the end-user and the quality of service requested or negotiated for the data session.

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

IpAppDataSessionRef

********** END OF FIRST CHANGE **********