

**3GPP TSG CN Plenary Meeting #22**  
**10 - 12 December 2003, Maui, Hawaii, USA**

**NP-030557**

**Source:** CN5 (OSA)  
**Title:** Rel-6 CRs 29.198-14 OSA API Part 14: Presence and Availability Management (PAM)  
**Agenda item:** 9.7  
**Document for:** APPROVAL

---

Doc-1st-Level	Spec	CR	R	Ph	Subject	Cat	Version-Current	Doc-2nd-Lev	WI
NP-030557	29.198-14	015	-	Rel-6	Include Provisioning SCF in Presence Service - Align with SA1's 22.141	B	5.3.0	N5-030628	OSA3
NP-030557	29.198-14	016	-	Rel-6	Add methods to activate/deactivate PAM services for users - Align with SA1's 22.141	B	5.3.0	N5-030629	OSA3

## CHANGE REQUEST

⌘ **29.198-14 CR 015** ⌘ rev - ⌘ Current version: **5.3.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Include Provisioning SCF in Presence Service - Align with SA1's 22.141		
<b>Source:</b>	⌘ CN5 (Teltier, ETSI PTCC)		
<b>Work item code:</b>	⌘ OSA3	<b>Date:</b>	⌘ 31/10/2003
<b>Category:</b>	⌘ <b>B</b>	<b>Release:</b>	⌘ REL-6
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="#">TR 21.900</a> .		Use <u>one</u> 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) Rel-6 (Release 6)

<b>Reason for change:</b>	⌘ Addition of feature to satisfy 3GPP Presence service requirement. The requirements are in clause 9 of 3GPP Presence requirements in TS 22.141.		
<b>Summary of change:</b>	⌘ Provisioning SCF added to Presence Service		
<b>Consequences if not approved:</b>	⌘ 3GPP Presence service requirements will not be satisfied in Release 6.		

<b>Clauses affected:</b>	⌘ 6, 8										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> <tr> <td style="text-align: center;"> </td> <td style="text-align: center;">X</td> </tr> </table>	Y	N		X		X		X	Other core specifications Test specifications O&M Specifications	⌘
Y	N										
	X										
	X										
	X										
<b>Other comments:</b>	⌘										

**How to create CRs using this form:**

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.htm>.

---

## 6 Class Diagrams

PAM consists of the following SCFs:

- [PAM Provisioning Service consisting of Interfaces to provision identities, agents and relationships between them](#)
- PAM AccessService consisting of interfaces to view and update presence and availability information and
- PAM EventManagement Service consisting of interfaces to subscribe to events in PAM and be notified of such events.

### [6.1 PAM Provisioning SCF Class Diagrams](#)

[PAM Provisioning Service contains the following service interfaces. It contains no application interfaces.](#)

#### [IpPAMIdentityManagement](#)

[- The purpose of this interface is to manage subscriber names, aliases, groups and sets of attributes called Profiles associated with subscribers.](#)

#### [IpPAMAgentManagement](#)

[- The purpose of this interface is to manage agent \(that models a hardware or software device\) names, communication capabilities and sets of attributes called Profiles associated with agents.](#)

#### [IpPAMAgentAssignment](#)

[- The purpose of this interface is to manage the relationship between identities and the agents assigned to them.](#)

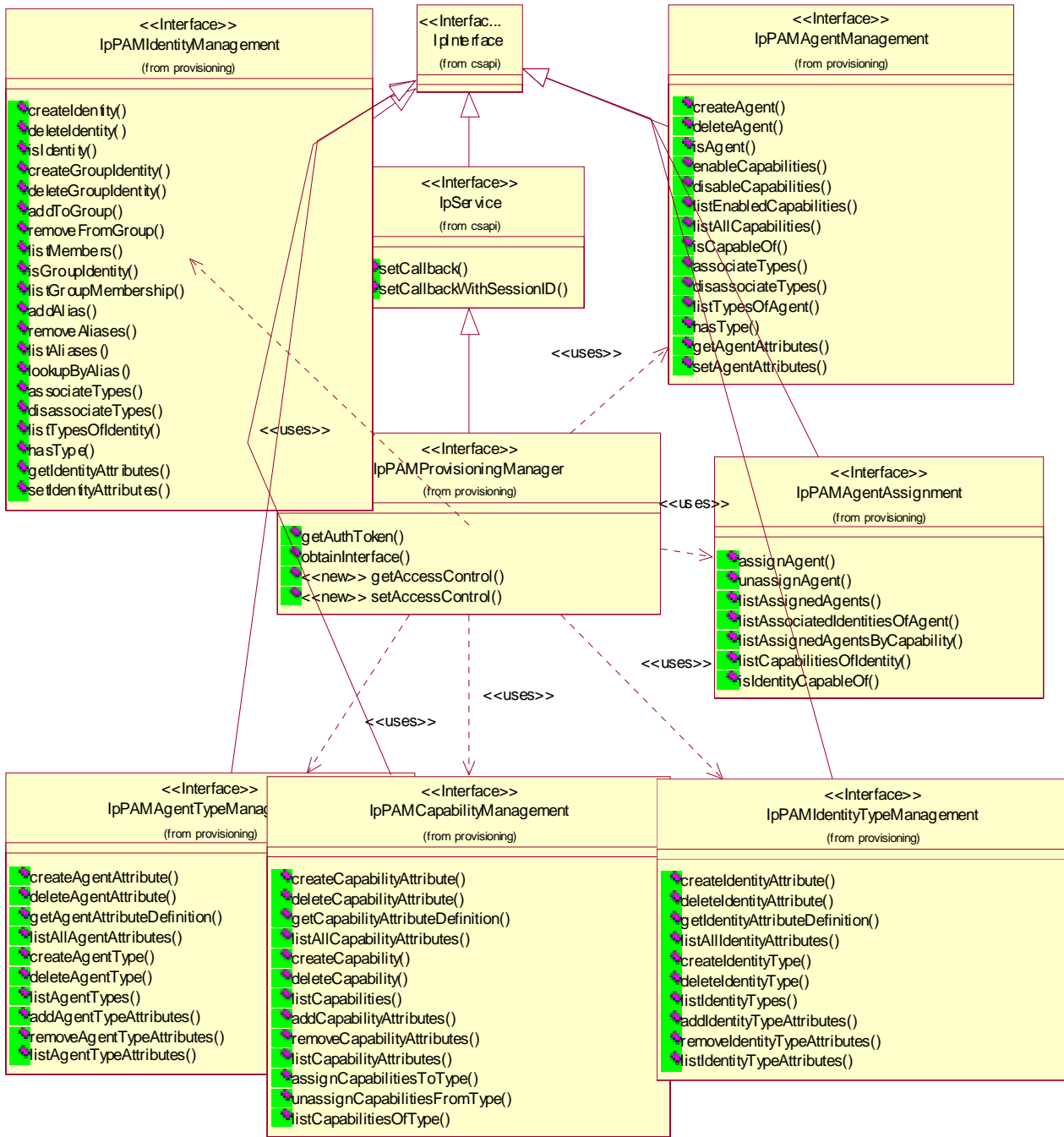
#### [IpPAMIdentityTypeManagement](#)

#### [IpPAMAgentTypeManagement](#)

#### [IpPAMCapabilityManagement](#)

[- These three interfaces allow for definitions of identity types, agent types and agent capabilities used for presence and availability management.](#)

[The interfaces and the relationships between them are shown in the figure below.](#)



**Figure 1: PAM Provisioning Service**

**6.16.2 PAM Access SCF Class Diagrams**

**6.26.3 PAM Event SCF Class Diagrams**

---

## 8 Presence and Availability Management Interface Classes

PAM consists of the following SCFs

- PAM Provisioning Service (~~not included in the 3GPP release 5 specifications~~)
- PAM Access Service
- PAM Event Service

The PAM Provisioning service consists of the Identity Management, Agent Management, Agent Assignment, Agent Type Management, Identity Type Management and Capability Management interfaces.

The PAM Access service consists of the identity presence and availability interfaces.

The Event service consists of the Event Management interfaces.

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.

### 8.1 PAM Provisioning SCF Interface Classes

This service consists of interfaces to provision identities, agents and the relationships between them.

#### 8.1.1 Interface Class IpPAMProvisioningManager

Inherits from: IpService.

The purpose of this interface is to supply the various interfaces available in this service to the application and to provide the authentication credentials. This interface is the only discoverable interface from the framework.

All PAM methods use an authentication token as a parameter since the outcome of the operations may depend on the entity requesting the operation. To enable this, the getAuthToken() method is used to obtain an implementation dependent token. An application that has authenticated itself with the OSA framework, can get an authentication token for itself. Alternatively, if the application is requesting PAM operations on behalf of multiple entities, authentication tokens may be requested for each such entity after providing any available data about the asker. These tokens can then be used repeatedly for operations within a session without further need to identify the asker.

<<Interface>>

IpPAMProvisioningManager

getAuthToken (askerData : in TpAttributeList) : TpPAMCredential

obtainInterface (interfaceName : in TpPAMProvisioningInterfaceName) : IpInterfaceRef

getAccessControl (identity : in TpPAMFQName, authToken : in TpPAMCredential) :  
TpPAMAccessControlData

setAccessControl (identity : in TpPAMFQName, operation : in TpPAMPreferenceOp, newAccessControl : in  
TpPAMAccessControlData, authToken : in TpPAMCredential) : void

### 8.1.1.1 Method getAuthToken()

Get an authentication token for access to the interface methods.

Returns an implementation-dependent authentication credential that can be verified.

#### Parameters

**askerData : in TpAttributeList**

Specifies information about the asker. Can be an empty array. The exact attributes in this list are dependent on the application. PAM reserves the attribute "name" with type TpPAMFQName to contain the identity of the asker if known.

#### Returns

**TpPAMCredential**

#### Raises

**TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.1.2 Method obtainInterface()

Obtain available interfaces from the service. The valid parameters for this method can be obtained from the service property P\_OBTAINABLE\_INTERFACES.

Returns the requested interface.

#### Parameters

**interfaceName : in TpPAMProvisioningInterfaceName**

Specifies the name of the required interface.

### Returns

[IpInterfaceRef](#)

### Raises

[TpCommonExceptions, P\\_PAM\\_UNAVAILABLE\\_INTERFACE](#)

#### 8.1.1.3 Method [getAccessControl\(\)](#)

[Get the access control associated with the data belonging to an identity. The data associated with an identity includes the static and dynamic attributes of an identity as well as data about agents associated with an identity.](#)

[This method should be used in conjunction with the \[setAccessControl\]\(#\) method.](#)

[Returns the access control if previously specified for the identity. Is null if there is no access control associated.](#)

### Parameters

[identity : in \[TpPAMFQName\]\(#\)](#)

[Specifies the identity of interest.](#)

[authToken : in \[TpPAMCredential\]\(#\)](#)

[Of the entity who wishes to do this operation.](#)

### Returns

[TpPAMAccessControlData](#)

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

#### 8.1.1.4 Method [setAccessControl\(\)](#)

[Set the access controls for the data associated with the specified identity. If the identity is Null, the access control is set for all identities \(if authorized to do so\). The data associated with an identity includes the static and dynamic attributes of an identity as well as data about agents associated with an identity.](#)

[Any existing access control will be modified based on the operation.](#)

[If the new access control is specified as Null for replace operation , an existing access control will be removed.](#)

### Parameters

[identity : in \[TpPAMFQName\]\(#\)](#)

[Specifies the identity of interest.](#)

[operation : in \[TpPAMPreferenceOp\]\(#\)](#)

[Specifies the operation to be performed with the specified preference.](#)

[newAccessControl : in \[TpPAMAccessControlData\]\(#\)](#)

[Specifies the access controls to add.](#)

[authToken : in \[TpPAMCredential\]\(#\)](#)

[Of the entity who wishes to do this operation.](#)

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL

## 8.1.2 Interface Class IpPAMIdentityManagement

Inherits from: IpInterface.

The purpose of this interface is to manage end-user or entity names, aliases, groups and sets of attributes associated with identities. An implementation may map these methods to operations on existing directories or databases. Some implementations may choose to provide a read-only access to the identity information.

The names of identities within a namespace must be unique. Each implementation exports an identifier as the default namespace that it serves. The identity name and the namespace may be used as an alias to another identity in a different namespace.

Aliases are associated with a given identity or group identity. Aliases must be uniquely assigned. In other words, two identities may not share the same alias.

This interface is meant for use by provisioning applications that establish and maintain identity names.



<<Interface>>

IpPAMIdentityManagement

createIdentity (identity : in TpPAMFQName, identityTypes : in TpStringList, authToken : in TpPAMCredential) : void

deleteIdentity (identity : in TpPAMFQName, authToken : in TpPAMCredential) : void

isIdentity (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpBoolean

createGroupIdentity (identity : in TpPAMFQName, identityTypes : in TpStringList, authToken : in TpPAMCredential) : void

deleteGroupIdentity (identity : in TpPAMFQName, authToken : in TpPAMCredential) : void

addToGroup (group : in TpPAMFQName, member : in TpPAMFQName, authToken : in TpPAMCredential) : void

removeFromGroup (group : in TpPAMFQName, identity : in TpPAMFQName, authToken : in TpPAMCredential) : void

listMembers (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMFQNameList

isGroupIdentity (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpBoolean

listGroupMembership (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMFQNameList

addAlias (identity : in TpPAMFQName, alias : in TpPAMFQName, authToken : in TpPAMCredential) : void

removeAliases (identity : in TpPAMFQName, alias : in TpPAMFQName, authToken : in TpPAMCredential) : void

listAliases (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMFQNameList

lookupByAlias (alias : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMFQName

associateTypes (identity : in TpPAMFQName, identityTypes : in TpStringList, authToken : in TpPAMCredential) : void

disassociateTypes (identity : in TpPAMFQName, identityTypes : in TpStringList, authToken : in TpPAMCredential) : void

listTypesOfIdentity (identity : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMFQNameList

hasType (identity : in TpPAMFQName, typeName : in TpString, authToken : in TpPAMCredential) : TpBoolean

getIdentityAttributes (identity : in TpPAMFQName, identityType : in TpString, attributeNames : in TpStringList, authToken : in TpPAMCredential) : TpPAMAttributeList

setIdentityAttributes (identity : in TpPAMFQName, identityType : in TpString, attributes : in TpPAMAttributeList, authToken : in TpPAMCredential) : void

### 8.1.2.1 Method createIdentity()

Create a new non-Group Identity with the specified name. Names must be unique across both group identities and non-group identities. Names must be unique across types within a namespace.

#### Parameters

**identity : in TpPAMFQName**

Specifies the Identity to be created.

**identityTypes : in TpStringList**

Specifies the identity's associated types. Can be an empty array.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Raises*

**TpCommonExceptions, P\_PAM\_IDENTITY\_EXISTS, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

**8.1.2.2 Method deleteIdentity()**

Delete the specified identity and all its related data. Upon successful completion, associated aliases and attribute instances are deleted from the system. The identity is also removed from all groups of which the identity is a member.

*Parameters*

**identity : in TpPAMFQName**

Specifies the Identity to be deleted.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

**8.1.2.3 Method isIdentity()**

Check if the specified non-group Identity exists. The method returns false for group identities.

Returns true if an identity with the specified name exists and false otherwise.

*Parameters*

**identity : in TpPAMFQName**

Specifies the Identity to be checked.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Returns*

**TpBoolean**

*Raises*

**TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL**

#### 8.1.2.4 Method createGroupIdentity()

Create a new Group Identity with the specified name. Name must be unique across both group identities and non-group identities. Names must be unique across the same types.

##### Parameters

**identity : in TpPAMFQName**

Specifies the group Identity to be created.

**identityTypes : in TpStringList**

Specifies the group's associated types. Can be an empty array.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

##### Raises

**TpCommonExceptions, P\_PAM\_IDENTITY\_EXISTS, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

#### 8.1.2.5 Method deleteGroupIdentity()

Delete the specified group identity and all its related data. Upon successful completion, associated aliases and attribute instances are deleted from the system. The identity is also removed from all groups of which the identity is a member. The member identities of the group are not deleted.

##### Parameters

**identity : in TpPAMFQName**

Specifies the group Identity to be deleted.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

##### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

#### 8.1.2.6 Method addToGroup()

Add an existing identity to a group identity. Both the group identity and the member identity to be added must have been created before this operation can be invoked. A member identity can be a group identity. Implementation must not allow cycles in memberships.

##### Parameters

**group : in TpPAMFQName**

Specifies the group Identity to which the member will be added.

**member : in TpPAMFQName**

Specifies the identity to be added as a member of the group.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Raises*

TpCommonExceptions, P\_PAM\_MEMBER\_EXISTS, P\_PAM\_UNKNOWN\_GROUP, P\_PAM\_UNKNOWN\_MEMBER, P\_PAM\_IS\_CYCLIC, P\_PAM\_INVALID\_CREDENTIAL

**8.1.2.7 Method removeFromGroup()**

Remove an existing identity from the membership of a group identity.

*Parameters*

**group : in TpPAMFQName**

Specifies the Group Identity from which the member will be removed.

**identity : in TpPAMFQName**

Specifies the Identity to be removed as a member of the group.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Raises*

TpCommonExceptions, P\_PAM\_NOT\_MEMBER, P\_PAM\_UNKNOWN\_GROUP, P\_PAM\_UNKNOWN\_MEMBER, P\_PAM\_INVALID\_CREDENTIAL

**8.1.2.8 Method listMembers()**

List the members of the specified group Identity.

Returns a list of members of the specified group Identity. An empty list is returned if the identity has no members.

*Parameters*

**identity : in TpPAMFQName**

Specifies the group Identity whose members are required.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

Returns

TpPAMFQNameList

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_GROUP, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.2.9 Method isGroupIdentity()

Check if the specified group identity exists. The method returns false for non-group identities

Returns true if a group identity with the specified name exists, false otherwise.

Parameters

identity : in TpPAMFQName

Specifies the Identity to be checked.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Returns

TpBoolean

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.2.10 Method listGroupMembership()

List the Group Identities the specified Identity is a member of.

Returns a list of all groups the specified Identity is member of. An empty list is returned if the identity is not member of any group.

Parameters

identity : in TpPAMFQName

Specifies the Identity to be looked up.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

### Returns

[TpPAMFQNameList](#)

### Raises

[TpCommonExceptions](#), [P\\_PAM\\_UNKNOWN\\_IDENTITY](#), [P\\_PAM\\_INVALID\\_CREDENTIAL](#)

## 8.1.2.11 Method addAlias()

Add an alias in the specified namespace to an existing Identity. The alias domain name must be specified in Alias parameter. The identity can be a group identity.

### Parameters

**identity : in [TpPAMFQName](#)**

Specifies the Identity to which the alias will be added.

**alias : in [TpPAMFQName](#)**

Specifies the alias to be added.

**authToken : in [TpPAMCredential](#)**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions](#), [P\\_PAM\\_ALIAS\\_EXISTS](#), [P\\_PAM\\_ALIAS\\_NOT\\_UNIQUE](#), [P\\_PAM\\_UNKNOWN\\_IDENTITY](#), [P\\_PAM\\_INVALID\\_CREDENTIAL](#)

## 8.1.2.12 Method removeAliases()

Remove the specified alias from an existing identity.

### Parameters

**identity : in [TpPAMFQName](#)**

Specifies the Identity from which the alias will be deleted.

**alias : in [TpPAMFQName](#)**

Specifies the alias to be deleted.

**authToken : in [TpPAMCredential](#)**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions](#), [P\\_PAM\\_UNKNOWN\\_IDENTITY](#), [P\\_PAM\\_UNASSIGNED\\_ALIAS](#), [P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.2.13 Method listAliases()

List the aliases of the specified Identity.

Returns a list containing all aliases to the specified Identity. An list with zero elements is returned if there are no aliases associated with the identity.

#### Parameters

**identity : in TpPAMFQName**

Specifies the Identity to be looked up.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### Returns

**TpPAMFQNameList**

#### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.2.14 Method lookupByAlias()

Find the identity with the specified alias in the specified alias domain.

Returns the identity that has the specified alias. Returns null if the alias is not assigned to any identity.

#### Parameters

**alias : in TpPAMFQName**

Specifies the alias to be looked up.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### Returns

**TpPAMFQName**

#### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_ALIAS, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.2.15 Method associateTypes()

Associate an identity instance with the specified types. The identity will be associated with instances of any attributes defined with each type. The initial values of the attributes will be as specified in the definition of the type attributes.

Parameters

identity : in TpPAMFQName

Specifies the name of the identity

identityTypes : in TpStringList

Specifies the names of the type to be associated.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_TYPE\_ASSOCIATED, P\_PAM\_INVALID\_CREDENTIAL

8.1.2.16 Method disassociateTypes()

Remove the association of a type with an identity instance. The definition of the type itself remains unaffected and the types may continue to be associated with other identities.

Parameters

identity : in TpPAMFQName

Specifies the identity.

identityTypes : in TpStringList

Specifies the names of the types to be removed.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_DISASSOCIATED\_TYPE, P\_PAM\_INVALID\_CREDENTIAL

8.1.2.17 Method listTypesOfIdentity()

List the types associated with the specified Identity.

Returns a list containing all types associated with the specified Identity. A list with zero elements is returned if there are no types associated with the identity.

Parameters

identity : in TpPAMFQName

Specifies the Identity to be looked up.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.



Returns

TpPAMFQNameList

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.2.18 Method hasType()

Check if the specified identity has the named type associated with it.

Returns true if an identity with the specified name has the named type associated with it, false otherwise.

Parameters

identity : in TpPAMFQName

Specifies the Identity to be checked.

typeName : in TpString

Specifies the type to be checked for.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Returns

TpBoolean

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.2.19 Method getIdentityAttributes()

Return the attributes associated with the identity. If the identity type is not specified, all associated types are assumed to be of interest.

The return value contains the list of specified attributes and their values. If the attributes parameter is an empty array, all attributes in the named identity are output.

Parameters

identity : in TpPAMFQName

Specifies the Identity whose attributes are to be accessed.

identityType : in TpString

Specifies the type of the identity with which the required attributes are associated. Is optional.

attributeNames : in TpStringList

List of attributes of interest. Can be an empty array.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Returns*

**TpPAMAttributeList**

*Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL**

### **8.1.2.20 Method setIdentityAttributes()**

Modify the attributes associated with the named Identity. The input may contain a subset of the attributes of the named type. Only the specified attributes will be modified and the rest will remain unchanged. If the type is unspecified, any associated type will be assumed

*Parameters*

**identity : in TpPAMFQName**

Specifies the Identity.

**identityType : in TpString**

Specifies the type of the identity for the operation. Is optional.

**attributes : in TpPAMAttributeList**

Contains the list of attributes and their values.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

*Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTES, P\_PAM\_INVALID\_CREDENTIAL**

## **8.1.3 Interface Class IpPAMAgentManagement**

Inherits from: IpInterface.

The purpose of this interface is to manage agent (that models a hardware or software device) names, communication capabilities and sets of attributes associated with agents. An implementation may map these methods to operations on existing directories or databases. Some implementations may choose to provide a read-only access to the agent information.

Data associated with an agent is captured in attributes associated with types. An implementation may map different type attributes to different underlying stores or directories.

The names of agents within a namespace must be unique.

This interface is meant for use by provisioning applications that establish and maintain agent names.

<<Interface>>

IpPAMAgentManagement

createAgent (agentName : in TpPAMFQName, agentTypes : in TpStringList, authToken : in TpPAMCredential) : void

deleteAgent (agentName : in TpPAMFQName, authToken : in TpPAMCredential) : void

isAgent (agentName : in TpPAMFQName, authToken : in TpPAMCredential) : TpBoolean

enableCapabilities (agentName : in TpPAMFQName, capabilities : in TpPAMCapabilityList, authToken : in TpPAMCredential) : void

disableCapabilities (agentName : in TpPAMFQName, capabilities : in TpPAMCapabilityList, authToken : in TpPAMCredential) : void

listEnabledCapabilities (agentName : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMCapabilityList

listAllCapabilities (agentName : in TpPAMFQName, authToken : in TpPAMCredential) : TpPAMCapabilityList

isCapableOf (agentName : in TpPAMFQName, capability : in TpPAMCapability, authToken : in TpPAMCredential) : TpBoolean

associateTypes (agentName : in TpPAMFQName, agentTypes : in TpStringList, authToken : in TpPAMCredential) : void

disassociateTypes (agentName : in TpPAMFQName, agentTypes : in TpStringList, authToken : in TpPAMCredential) : void

listTypesOfAgent (agentName : in TpPAMFQName, authToken : in TpPAMCredential) : TpStringList

hasType (agentName : in TpPAMFQName, typeName : in TpString, authToken : in TpPAMCredential) : TpBoolean

getAgentAttributes (agentName : in TpPAMFQName, agentType : in TpString, attributeNames : in TpStringList, authToken : in TpPAMCredential) : TpPAMAttributeList

setAgentAttributes (agentName : in TpPAMFQName, agentType : in TpString, attributes : in TpPAMAttributeList, authToken : in TpPAMCredential) : void

### 8.1.3.1 Method createAgent()

Create an agent initialized with the specified capabilities.

#### *Parameters*

**agentName : in TpPAMFQName**

Specifies the name of Agent to be created.

**agentTypes : in TpStringList**

Specifies the types of the Agent to be created. Can be an empty list.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_AGENT\\_EXISTS, P\\_PAM\\_UNKNOWN\\_TYPE, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.3.2 Method deleteAgent()

Delete the specified Agent and all related data from the system.

### Parameters

**agentName : in TpPAMFQName**

Specifies the name of Agent to be created.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_AGENT, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.3.3 Method isAgent()

Check if the specified agent exists.

Returns true if an agent with the specified name exists, false otherwise.

### Parameters

**agentName : in TpPAMFQName**

Specifies the Agent to be checked.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Returns

**TpBoolean**

### Raises

[TpCommonExceptions, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.3.4 Method enableCapabilities()

Enable the specified capabilities of the agent.

Parameters

agentName : in TpPAMFQName

Specifies the Agent.

capabilities : in TpPAMCapabilityList

Specifies the capabilities to be enabled.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL

8.1.3.5 Method disableCapabilities()

Remove the specified capabilities from the Agent.

Parameters

agentName : in TpPAMFQName

Specifies the Agent.

capabilities : in TpPAMCapabilityList

Specifies the communication mode to be disabled.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_NO\_CAPABILITY, P\_PAM\_INVALID\_CREDENTIAL

8.1.3.6 Method listEnabledCapabilities()

List the enabled capabilities for the specified Agent.

Returns the list of enabled capabilities for the Agent. Returns an empty list if no enabled capabilities exist for the agent.

Parameters

agentName : in TpPAMFQName

Specifies the Agent whose capabilities are to be listed.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

Returns

**TpPAMCapabilityList**

Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

8.1.3.7 Method listAllCapabilities()

List the capabilities for the specified Agent.

Returns the list of capabilities for the Agent. Returns an empty list if no capabilities exist for the agent.

Parameters

**agentName : in TpPAMFQName**

Specifies the Agent whose capabilities are to be listed.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

Returns

**TpPAMCapabilityList**

Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

8.1.3.8 Method isCapableOf()

Check if an agent has a particular capability that is currently enabled.

Returns true if the agent has the specified capability, false otherwise.

Parameters

**agentName : in TpPAMFQName**

Specifies the Agent to be checked.

**capability : in TpPAMCapability**

Capability to be checked.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Returns

**TpBoolean**

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.3.9 Method associateTypes()

Associate an agent instance with the specified types.

### Parameters

**agentName : in TpPAMFQName**

Specifies the name of Agent.

**agentTypes : in TpStringList**

Specifies the types of the Agent to be associated.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_TYPE\_ASSOCIATED, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.3.10 Method disassociateTypes()

Remove the association of a type with an agent instance. The definition of the type itself remains unaffected and the types may continue to be associated with other agents.

### Parameters

**agentName : in TpPAMFQName**

Specifies the agent.

**agentTypes : in TpStringList**

Specifies the names of the types to be removed.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_DISASSOCIATED\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.3.11 Method listTypesOfAgent()

List the types associated with the specified agent.

Returns the list containing all types associated with the specified agent. An empty list is returned if there are no types associated with the agent.

#### *Parameters*

**agentName : in TpPAMFQName**

Specifies the agent to be looked up.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### *Returns*

**TpStringList**

#### *Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.3.12 Method hasType()

Check if the specified agent has the named type associated with it.

Returns true if an agent with the specified name has the named type associated with it.

#### *Parameters*

**agentName : in TpPAMFQName**

Specifies the Agent to be checked.

**typeName : in TpString**

Specifies the type to be checked for.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### *Returns*

**TpBoolean**

#### *Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.3.13 Method getAgentAttributes()

Return the attributes associated with the agent. If the type is not specified, all associated types are assumed.



Return value contains the list of specified attributes and their values. If the attributes parameter is an empty list, all attributes in the named agent are returned.

#### Parameters

agentName : in TpPAMFQName

Specifies the agent.

agentType : in TpString

Specifies the type of interest. Is optional.

attributeNames : in TpStringList

List of attributes of interest. Can be an empty list.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

#### Returns

TpPAMAttributeList

#### Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.3.14 Method setAgentAttributes()

Modify the attributes associated with the named agent. The input may contain a subset of the attributes of the named type. Only the specified attributes will be modified and the rest will remain unchanged.

#### Parameters

agentName : in TpPAMFQName

Specifies the agent.

agentType : in TpString

Specifies the type of interest. Is optional.

attributes : in TpPAMAttributeList

Contains the list of attributes and their values.

authToken : in TpPAMCredential

Of the entity who wishes to do this operation.

#### Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTES, P\_PAM\_INVALID\_CREDENTIAL

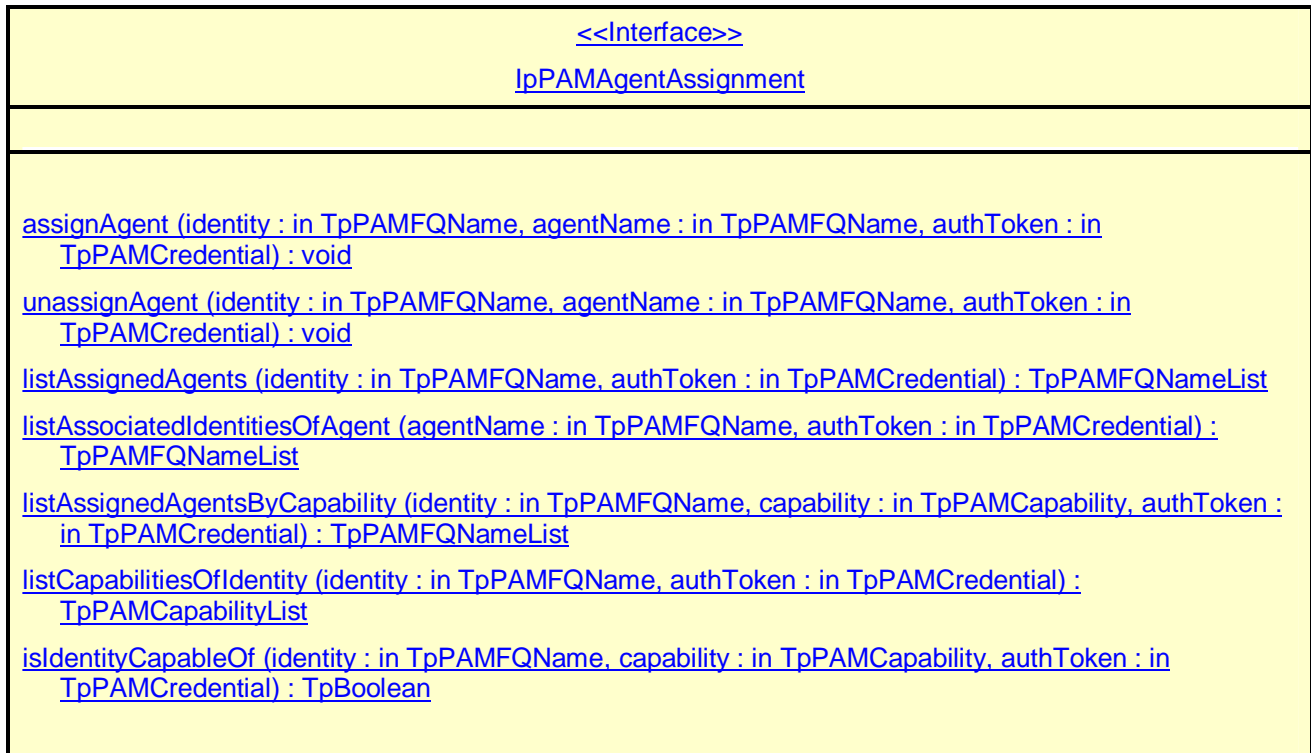
## 8.1.4 Interface Class IpPAMAgentAssignment

Inherits from: IpInterface.

The purpose of this interface is to manage the relationship between identities and the agents assigned to them. The identities inherit capabilities from the assignments of agents.

The implementation must maintain the integrity of the relationship between identities and agents across changes to both identities and agents. Implementations may map these methods to operations on existing directories and databases. Some implementations may provide a read-only access to this interface.

This interface is meant for use by provisioning applications that establish and maintain association of agents with identities.



### 8.1.4.1 Method assignAgent()

Assign an existing agent to an existing identity.

#### Parameters

**identity : in TpPAMFQName**

Specifies the identity to assign the agent to.

**agentName : in TpPAMFQName**

Specifies the Agent.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_UNKNOWN\\_AGENT, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.4.2 Method unassignAgent()

Unassign an agent from an existing identity. In effect, this deletes an existing relationship between an agent and an identity.

### Parameters

**identity : in TpPAMFQName**

Specifies the identity to assign the agent to.

**agentName : in TpPAMFQName**

Specifies the Agent.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_UNKNOWN\\_AGENT, P\\_PAM\\_UNKNOWN\\_ASSIGNMENT, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.4.3 Method listAssignedAgents()

List the Agents assigned to an identity.

Returns the list of agent names assigned to the identity. An empty list is returned if no agents are assigned to the identity.

### Parameters

**identity : in TpPAMFQName**

Specifies the identity of interest.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Returns

**TpPAMFQNameList**

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

#### 8.1.4.4 Method listAssociatedIdentitiesOfAgent()

List the identities that have the specified agent assigned to them.

Returns the list of identities that have been assigned the specified agent. Empty list is returned if no identities have been assigned this agent.

##### *Parameters*

**agentName : in TpPAMFQName**

Specifies the Agent.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

##### *Returns*

**TpPAMFQNameList**

##### *Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_AGENT, P\_PAM\_INVALID\_CREDENTIAL**

#### 8.1.4.5 Method listAssignedAgentsByCapability()

List the Agents assigned to an identity that match the specified capability.

Returns the list of agent names with the specified capability. An empty list is returned no agents are found.

##### *Parameters*

**identity : in TpPAMFQName**

Specifies the identity of interest.

**capability : in TpPAMCapability**

Specifies the capability of interest.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

##### *Returns*

**TpPAMFQNameList**

##### *Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

#### 8.1.4.6 Method listCapabilitiesOfIdentity()

List the capabilities of an identity that it derives from its assigned Agents.

Returns the list of the identity's capabilities. Returns an empty list if no capabilities exist.

### Parameters

**identity : in TpPAMFQName**

Specifies the identity of interest.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Returns

**TpPAMCapabilityList**

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.4.7 Method isIdentityCapableOf()

Check if an identity has the specified capability derived from one or more agents assigned to it.

Returns true if the identity has this capability, false otherwise.

### Parameters

**identity : in TpPAMFQName**

Specifies the identity of interest.

**capability : in TpPAMCapability**

Identifies the capability to check for.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

### Returns

**TpBoolean**

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.5 Interface Class IpPAMIdentityTypeManagement

Inherits from: IpInterface.

This clause describes the programmatic interface to define the type schema for identities specifying the attributes associated with the type. These types can then be assigned to identities. PAM implementations may provide a set of pre-defined types. Identity type names and agent type names are in the same namespace and hence must be uniquely defined across both identities and agents. The attributes for identity types and agent types are in two different namespaces and hence an attribute name may be re-used with different characteristics for identities and agents.

<<Interface>>

IpPAMIdentityTypeManagement

createIdentityAttribute (pAttribute : in TpPAMAttributeDef, authToken : in TpPAMCredential) : void  
deleteIdentityAttribute (attributeName : in TpString, authToken : in TpPAMCredential) : void  
getIdentityAttributeDefinition (attributeName : in TpString, authToken : in TpPAMCredential) : TpPAMAttributeDef  
listAllIdentityAttributes (authToken : in TpPAMCredential) : TpStringList  
createIdentityType (typeName : in TpString, attributeNames : in TpStringList, authToken : in TpPAMCredential) : void  
deleteIdentityType (typeName : in TpString, authToken : in TpPAMCredential) : void  
listIdentityTypes (authToken : in TpPAMCredential) : TpStringList  
addIdentityTypeAttributes (typeName : in TpString, attributeNames : in TpStringList, authToken : in TpPAMCredential) : void  
removeIdentityTypeAttributes (typeName : in TpString, attributeNames : in TpStringList, authToken : in TpPAMCredential) : void  
listIdentityTypeAttributes (typeName : in TpString, authToken : in TpPAMCredential) : TpStringList

8.1.5.1 Method createIdentityAttribute()

Create a definition of an identity attribute to specify its name and type.

Parameters

**pAttribute : in TpPAMAttributeDef**

Specifies the attribute to be created.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Raises

**TpCommonExceptions, P\_PAM\_ATTRIBUTE\_EXISTS, P\_PAM\_INVALID\_CREDENTIAL**

8.1.5.2 Method deleteIdentityAttribute()

Delete the definition of an identity attribute.

Parameters

**attributeName : in TpString**

Specifies the attribute to be deleted.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.5.3 Method getIdentityAttributeDefinition()

Get the definition for the specified identity attribute.

Returns the definition of the specified attribute.

Parameters

attributeName : in TpString

Specifies the attribute.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Returns

TpPAMAttributeDef

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.5.4 Method listAllIdentityAttributes()

List all known identity attributes.

Returns the list of attribute names defined so far. An empty array if no attributes have been defined.

Parameters

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Returns

TpStringList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

8.1.5.5 Method createIdentityType()

Specify a label as the name of an identity type.

### Parameters

**typeName : in TpString**

Specifies the name of the type to be created.

**attributeNames : in TpStringList**

Specifies the list of attributes to be added.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_TYPE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.5.6 Method deleteIdentityType()

Delete a label as the name of an identity type. All identities that have this type are no longer associated with this type and consequently will no longer will have any attributes associated with this type.

### Parameters

**typeName : in TpString**

Specifies the name of the type.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.5.7 Method listIdentityTypes()

List all known identity types.

Returns the list of known identity types. An empty array if no identity types have been defined.

### Parameters

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.



Returns

TpStringList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.5.8 Method addIdentityTypeAttributes()

Add attribute definitions to the schema of an identity type that has already been defined.

Parameters

typeName : in TpString

Specifies the name of the type.

attributeNames : in TpStringList

List of attributes to be added to this type.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_ATTRIBUTE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.5.9 Method removeIdentityTypeAttributes()

Delete attribute definitions from the schema of an identity type that has already been defined.

Parameters

typeName : in TpString

Specifies the name of the type.

attributeNames : in TpStringList

Specifies the list of attributes to be deleted.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.5.10 Method listIdentityTypeAttributes()

List all attributes of an identity type.

Returns the list of attribute definitions for the named type. An empty array if no attributes have been defined for this type.

#### *Parameters*

**typeName : in TpString**

Specifies the name of the type.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

#### *Returns*

**TpStringList**

#### *Raises*

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.6 Interface Class IpPAMAgentTypeManagement

Inherits from: IpInterface.

This clause describes the programmatic interface to define the type schema for agents specifying the attributes associated with the type. These types can then be assigned to agents. PAM implementations may provide a set of pre-defined types. Identity type names and agent type names are in the same namespace and hence must be uniquely defined across both identities and agents. The attributes for identity types and agent types are in two different namespaces and hence an attribute name may be re-used with different characteristics for identities and agents.

<<Interface>>

IpPAMAgentTypeManagement

createAgentAttribute (pAttribute : in TpPAMAttributeDef, authToken : in TpPAMCredential) : void  
deleteAgentAttribute (attributeName : in TpString, authToken : in TpPAMCredential) : void  
getAgentAttributeDefinition (attributeName : in TpString, authToken : in TpPAMCredential) :  
TpPAMAttributeDef  
listAllAgentAttributes (authToken : in TpPAMCredential) : TpStringList  
createAgentType (typeName : in TpString, attributeNames : in TpStringList, authToken : in  
TpPAMCredential) : void  
deleteAgentType (typeName : in TpString, authToken : in TpPAMCredential) : void  
listAgentTypes (authToken : in TpPAMCredential) : TpStringList  
addAgentTypeAttributes (typeName : in TpString, attributeNames : in TpStringList, authToken : in  
TpPAMCredential) : void  
removeAgentTypeAttributes (typeName : in TpString, attributeNames : in TpStringList, authToken : in  
TpPAMCredential) : void  
listAgentTypeAttributes (typeName : in TpString, authToken : in TpPAMCredential) : TpStringList

8.1.6.1 Method createAgentAttribute()

Create a definition of an Agent attribute to specify its name and type.

Parameters

**pAttribute : in TpPAMAttributeDef**

Specifies the attribute to be created.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Raises

**TpCommonExceptions, P\_PAM\_ATTRIBUTE\_EXISTS, P\_PAM\_INVALID\_CREDENTIAL**

8.1.6.2 Method deleteAgentAttribute()

Delete the definition of an Agent attribute.

Parameters

**attributeName : in TpString**

Specifies the attribute to be deleted.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.6.3 Method getAgentAttributeDefinition()

Get the definition for the specified Agent attribute.

Returns the definition of the specified attribute.

Parameters

attributeName : in TpString

Specifies the attribute.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Returns

TpPAMAttributeDef

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.6.4 Method listAllAgentAttributes()

List all known Agent attributes.

Returns the list of attribute names defined so far. An empty array if no attributes have been defined.

Parameters

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Returns

TpStringList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

8.1.6.5 Method createAgentType()

Specify a label as the name of an Agent type.

### Parameters

**typeName : in TpString**

Specifies the name of the type to be created.

**attributeNames : in TpStringList**

Specifies the list of attributes to be added.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_TYPE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.6.6 Method deleteAgentType()

Delete a label as the name of an Agent type. All identities that have this type are no longer associated with this type and consequently will no longer will have any attributes associated with this type.

### Parameters

**typeName : in TpString**

Specifies the name of the type.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.6.7 Method listAgentTypes()

List all known Agent types.

Returns the list of known Agent types. An empty array if no Agent types have been defined.

### Parameters

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Returns

TpStringList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

8.1.6.8 Method addAgentTypeAttributes()

Add attribute definitions to the schema of an Agent type that has already been defined.

Parameters

typeName : in TpString

Specifies the name of the type.

attributeNames : in TpStringList

List of attributes to be added to this type.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_ATTRIBUTE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.6.9 Method removeAgentTypeAttributes()

Delete attribute definitions from the schema of an Agent type that has already been defined.

Parameters

typeName : in TpString

Specifies the name of the type.

attributeNames : in TpStringList

Specifies the list of attributes to be deleted.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.6.10 Method listAgentTypeAttributes()

List all attributes of an Agent type.

Returns the list of attributes for the named type. An empty array if no attributes have been defined for this type.

#### Parameters

**typeName : in TpString**

Specifies the name of the type.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

#### Returns

**TpStringList**

#### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.7 Interface Class IpPAMCapabilityManagement

Inherits from: IpInterface.

This clause describes the programmatic interface to define capability names.

Capabilities are names that define a property of an agent for which presence data may exist. Examples are voice, IM, SMS, WAP, etc. Agents can be assigned capabilities. Identities inherit capabilities from agents but cannot be directly assigned capabilities. Each capability is defined with an associated set of attributes. The attributes for each capability exist in their own namespace and hence an attribute name may be re-used with different characteristics across capabilities.

<<Interface>>

IpPAMCapabilityManagement

[createCapabilityAttribute \(pAttribute : in TpPAMAttributeDef, authToken : in TpPAMCredential\) : void](#)  
[deleteCapabilityAttribute \(attributeName : in TpString, authToken : in TpPAMCredential\) : void](#)  
[getCapabilityAttributeDefinition \(attributeName : in TpString, authToken : in TpPAMCredential\) : TpPAMAttributeDef](#)  
[listAllCapabilityAttributes \(authToken : in TpPAMCredential\) : TpStringList](#)  
[createCapability \(capabilityName : in TpPAMCapability, attributeNames : in TpStringList, authToken : in TpPAMCredential\) : void](#)  
[deleteCapability \(capabilityName : in TpPAMCapability, authToken : in TpPAMCredential\) : void](#)  
[listCapabilities \(authToken : in TpPAMCredential\) : TpPAMCapabilityList](#)  
[addCapabilityAttributes \(capabilityName : in TpPAMCapability, attributeNames : in TpStringList, authToken : in TpPAMCredential\) : void](#)  
[removeCapabilityAttributes \(capabilityName : in TpPAMCapability, attributeNames : in TpStringList, authToken : in TpPAMCredential\) : void](#)  
[listCapabilityAttributes \(capabilityName : in TpPAMCapability, authToken : in TpPAMCredential\) : TpStringList](#)  
[assignCapabilitiesToType \(agentType : in TpString, capabilities : in TpPAMCapabilityList, authToken : in TpPAMCredential\) : void](#)  
[unassignCapabilitiesFromType \(agentType : in TpString, capabilities : in TpPAMCapabilityList, authToken : in TpPAMCredential\) : void](#)  
[listCapabilitiesOfType \(agentType : in TpString, authToken : in TpPAMCredential\) : TpPAMCapabilityList](#)

[8.1.7.1 Method createCapabilityAttribute\(\)](#)

[Create a definition of a Capability attribute to specify its name and type.](#)

[Parameters](#)

[pAttribute : in TpPAMAttributeDef](#)

[Specifies the attribute to be created.](#)

[authToken : in TpPAMCredential](#)

[Credential of the entity who wishes to do this operation.](#)

[Raises](#)

[TpCommonExceptions, P\\_PAM\\_ATTRIBUTE\\_EXISTS, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

[8.1.7.2 Method deleteCapabilityAttribute\(\)](#)

[Delete the definition of a Capability attribute.](#)



### Parameters

**attributeName : in TpString**

Specifies the attribute to be deleted.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.7.3 Method getCapabilityAttributeDefinition()

Get the definition for the specified Capability attribute.

Returns the definition of the specified attribute.

### Parameters

**attributeName : in TpString**

Specifies the attribute.

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

### Returns

**TpPAMAttributeDef**

### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL**

## 8.1.7.4 Method listAllCapabilityAttributes()

List all known Capability attributes.

Returns the list of attribute names defined so far. An empty array if no attributes have been defined.

### Parameters

**authToken : in TpPAMCredential**

Credential of the entity who wishes to do this operation.

Returns

TpStringList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

8.1.7.5 Method createCapability()

Specify a label as the name of a Capability type.

Parameters

capabilityName : in TpPAMCapability

Specifies the name of the type to be created.

attributeNames : in TpStringList

Specifies the list of attributes to be added.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_TYPE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

8.1.7.6 Method deleteCapability()

Delete a label as the name of a Capability. All agents that have this type are no longer associated with this type and consequently will no longer will have any attributes associated with this type.

Parameters

capabilityName : in TpPAMCapability

Specifies the name of the type.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL

8.1.7.7 Method listCapabilities()

List all known Capability types.

Returns the list of known Capability types. An empty array if no Capability types have been defined.

Parameters

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Returns

TpPAMCapabilityList

Raises

TpCommonExceptions, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.7.8 Method addCapabilityAttributes()

Add attribute definitions to the schema of a capability that has already been defined.

Parameters

capabilityName : in TpPAMCapability

Specifies the name of the type.

attributeNames : in TpStringList

List of attributes to be added to this type.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_ATTRIBUTE\_EXISTS, P\_PAM\_UNKNOWN\_ATTRIBUTE, P\_PAM\_INVALID\_CREDENTIAL

### 8.1.7.9 Method removeCapabilityAttributes()

Delete attribute definitions from the schema of a Capability that has already been defined.

Parameters

capabilityName : in TpPAMCapability

Specifies the name of the type.

attributeNames : in TpStringList

Specifies the list of attributes to be deleted.

authToken : in TpPAMCredential

Credential of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_TYPE, P\\_PAM\\_UNKNOWN\\_ATTRIBUTE, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.7.10 Method listCapabilityAttributes()

List all attributes of a capability.

Returns the list of attributes for the named Capability. An empty array if no attributes have been defined for this capability.

### Parameters

[capabilityName : in TpPAMCapability](#)

Specifies the name of the type.

[authToken : in TpPAMCredential](#)

Credential of the entity who wishes to do this operation.

### Returns

[TpStringList](#)

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_TYPE, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.7.11 Method assignCapabilitiesToType()

Assign capabilities to agent type.

### Parameters

[agentType : in TpString](#)

Name of an agent type.

[capabilities : in TpPAMCapabilityList](#)

Specifies the list of capabilities to be associated.

[authToken : in TpPAMCredential](#)

Of the entity who wishes to do this operation.

### Raises

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_TYPE, P\\_PAM\\_UNKNOWN\\_CAPABILITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.1.7.12 Method unassignCapabilitiesFromType()

Unassign capabilities from an agent type.

#### Parameters

**agentType : in TpString**

Name of an agent type.

**capabilities : in TpPAMCapabilityList**

Specifies the list of capabilities to be disassociated.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_UNKNOWN\_CAPABILITY, P\_PAM\_INVALID\_CREDENTIAL**

### 8.1.7.13 Method listCapabilitiesOfType()

List capabilities assigned to an agent type.

Returns the list of capabilities assigned to the agent type.

#### Parameters

**agentType : in TpString**

Name of an agent type.

**authToken : in TpPAMCredential**

Of the entity who wishes to do this operation.

#### Returns

**TpPAMCapabilityList**

#### Raises

**TpCommonExceptions, P\_PAM\_UNKNOWN\_TYPE, P\_PAM\_INVALID\_CREDENTIAL**

## Annex C (informative): Change history

Change history							
Date	TSG #	TSG Doc.	CR	Rev	Subject/Comment	Old	New
April 2002	--	--	--	--	Draft v100 submitted to TSG CN email list for Information		1.0.0
June 2002	CN_16	NP-020196	--	--	Draft v200 submitted to TSG CN#16 for Approval	2.0.0	5.0.0
Sep 2002	CN_17	NP-020440	001	--	Add text to clarify requirements on support of methods	5.0.0	5.1.0
Sep 2002	CN_17	NP-020440	002	--	Remove declaration of unused datatype TpPAMTime	5.0.0	5.1.0
Sep 2002	CN_17	NP-020395	003	--	Add text to clarify relationship between 3GPP and ETSI/Parlay OSA specifications	5.0.0	5.1.0
Jun 2003	CN_20	NP-030245	004	--	Make TpPAMCapability extensible by changing its type to TpString	5.1.0	5.2.0
Jun 2003	CN_20	NP-030240	005	--	Change the type of TpPAMFQName to TpURN	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	006	--	Clarify use of askerData parameter to getAuthToken method in each PAM SCF	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	007	--	Add authToken parameter to computeAvailability method	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	008	--	Replace use of IpInterfaceRef in PAM with actual application interfaces	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	009	--	Add expiration time for PAM event registrations	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	010	--	Send subscription notification cancellation to watchers	5.1.0	5.2.0
Jun 2003	CN_20	NP-030241	011	--	Change PAM Presence and Availability SCF name to PAM Access	5.1.0	5.2.0
Jun 2003	CN_20	NP-030245	012	--	Move Access Control Mechanism to Manager Interface	5.1.0	5.2.0
Sep 2003	CN_21	NP-030352	013	--	Correction to Java Realisation Annex	5.2.0	5.3.0

## CHANGE REQUEST

⌘ **29.198-14 CR 016** ⌘ rev **-** ⌘ Current version: **5.3.0** ⌘

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

**Proposed change affects:** UICC apps  ME  Radio Access Network  Core Network

<b>Title:</b>	⌘ Add methods to activate/deactivate PAM services for users - Align with SA1's 22.141		
<b>Source:</b>	⌘ CN5 (Teltier, ETSI PTCC)		
<b>Work item code:</b>	⌘ OSA3	<b>Date:</b>	⌘ 31/10/2003
<b>Category:</b>	⌘ <b>B</b>	<b>Release:</b>	⌘ REL-6
	Use <u>one</u> of the following categories: <b>F</b> (correction) <b>A</b> (corresponds to a correction in an earlier release) <b>B</b> (addition of feature), <b>C</b> (functional modification of feature) <b>D</b> (editorial modification) Detailed explanations of the above categories can be found in 3GPP <a href="http://www.3gpp.org/Specs/tr21/900">TR 21.900</a> .		Use <u>one</u> 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) Rel-6 (Release 6)

<b>Reason for change:</b>	⌘ Addition of feature to satisfy 3GPP Presence service requirement. The requirements are in clauses 9.5 and 9.6 of 3GPP Presence requirements in TS 22.141.		
<b>Summary of change:</b>	⌘ Service activation methods added to top level managers for each SCF. Three new methods are introduced into each top level manager interfaces in sections 8.1.1 and 8.2.1. The signatures and semantics are the same in both the sections. The activation and deactivation affects only the SCF in which the methods are invoked which are Access and Event Management SCFs.		
<b>Consequences if not approved:</b>	⌘ 3GPP Presence service requirements will not be satisfied in Release 6.		

<b>Clauses affected:</b>	⌘ 8.1.1, 8.2.1										
<b>Other specs affected:</b>	<table border="1" style="display: inline-table; border-collapse: collapse;"> <tr> <td style="width: 20px; text-align: center;">Y</td> <td style="width: 20px; text-align: center;">N</td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> <tr> <td style="text-align: center;"><input type="checkbox"/></td> <td style="text-align: center;"><input checked="" type="checkbox"/></td> </tr> </table>	Y	N	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	Other core specifications ⌘ Test specifications ⌘ O&M Specifications ⌘	
Y	N										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<input type="checkbox"/>	<input checked="" type="checkbox"/>										
<b>Other comments:</b>	⌘										

**How to create CRs using this form:**

Comprehensive information and tips about how to create CRs can be found at <http://www.3gpp.org/specs/CR.htm>.

---

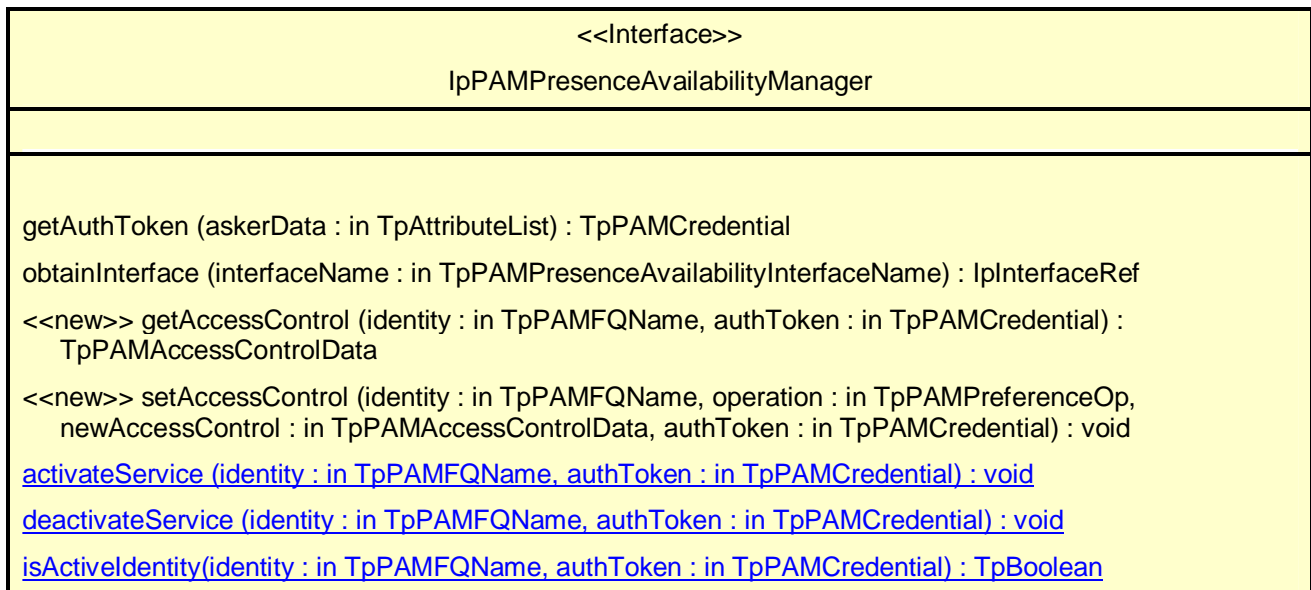
# Introduction

The current specification does not provide the means to activate/deactivate the service for specific users even though they have been provisioned into the service.

The contribution is a proposal to satisfy the 3GPP Presence requirements in TS 22.141 clauses 9.5 and 9.6.

←=====Changes in 8.1=====→

## 8.1.1 Interface Class IpPAMPresenceAvailabilityManager



### [8.1.1.5 Method activateService \(\)](#)

[Activate the service for the specified identity. If the identity is null, activate the service for ALL identities. The method succeeds if the service is already activated for the specified identity.](#)

#### *[Parameters](#)*

**[identity : in TpPAMFQName](#)**

[specifies the identity of interest. Can be NULL.](#)

**[authToken : in TpPAMCredential](#)**

[of the entity who wishes to do this operation.](#)

#### *[Raises](#)*

**[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)**

### [8.1.1.6 Method deactivateService \(\)](#)

[Deactivate the service for the specified identity. If the identity is null, deactivate the service for ALL identities. The method succeeds if the service is already deactivated for the specified identity.](#)



Parameters

identity : in TpPAMFQName

specifies the identity of interest. Can be NULL.

authToken : in TpPAMCredential

of the entity who wishes to do this operation.

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL

8.1.1.7 Method isActiveIdentity ()

Check if the service is active for the specified identity. Returns true if the service is active for that identity, false otherwise.

Parameters

identity : in TpPAMFQName

specifies the identity of interest. Cannot be NULL.

authToken : in TpPAMCredential

of the entity who wishes to do this operation.

Returns

TpBoolean

Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL

←=====Changes in 8.2=====→

8.2.1 Interface Class IpPAMEventManager

<<Interface>>  
IpPAMEventManager

getAuthToken (askerData : in TpAttributeList) : TpPAMCredential  
obtainInterface (interfaceName : in TpPAMEventInterfaceName) : IpInterfaceRef  
<<new>> getAccessControl (identity : in TpPAMFQName, authToken : in TpPAMCredential) :  
TpPAMAccessControlData  
<<new>> setAccessControl (identity : in TpPAMFQName, operation : in TpPAMPreferenceOp,  
newAccessControl : in TpPAMAccessControlData, authToken : in TpPAMCredential) : void  
[activateService \(identity : in TpPAMFQName, authToken : in TpPAMCredential\) : void](#)  
[deactivateService \(identity : in TpPAMFQName, authToken : in TpPAMCredential\) : void](#)  
[isActiveIdentity\(identity : in TpPAMFQName, authToken : in TpPAMCredential\) : TpBoolean](#)

### [8.2.1.5 Method activateService \(\)](#)

[Activate the service for the specified identity. If the identity is null, activate the service for ALL identities. The method succeeds if the service is already activated for the specified identity.](#)

#### [Parameters](#)

[identity : in TpPAMFQName](#)

[specifies the identity of interest. Can be NULL.](#)

[authToken : in TpPAMCredential](#)

[of the entity who wishes to do this operation.](#)

#### [Raises](#)

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### [8.2.1.6 Method deactivateService \(\)](#)

[Deactivate the service for the specified identity. If the identity is null, deactivate the service for ALL identities. The method succeeds if the service is already deactivated for the specified identity.](#)

#### [Parameters](#)

[identity : in TpPAMFQName](#)

[specifies the identity of interest. Can be NULL.](#)

[authToken : in TpPAMCredential](#)

[of the entity who wishes to do this operation.](#)

#### [Raises](#)

[TpCommonExceptions, P\\_PAM\\_UNKNOWN\\_IDENTITY, P\\_PAM\\_INVALID\\_CREDENTIAL](#)

### 8.2.1.7 Method isActiveIdentity ()

Check if the service is active for the specified identity. Returns true if the service is active for that identity, false otherwise.

#### Parameters

**identity : in TpPAMFQName**

specifies the identity of interest. Cannot be NULL.

**authToken : in TpPAMCredential**

of the entity who wishes to do this operation.

#### Returns

TpBoolean

#### Raises

TpCommonExceptions, P\_PAM\_UNKNOWN\_IDENTITY, P\_PAM\_INVALID\_CREDENTIAL