PAM Specification Document

Version: 1.0

Date: Sept 11, 2001

This is a public release of the Presence and Availability Management Specification from the PAMforum.

Any feedback or comments can be sent to feedback@pamforum.org

For more information about the PAMforum, please visit http://www.pamforum.org

Copyright Notice

Intellectual Property Rights statement

The PAMforum takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the PAMforum Executive Director.

1 Introduction

This document defines and proposes, for industry adoption, a set of specifications for Presence and Availability Management (PAM). The goal is to establish, through industry consensus and adoption, a standard for maintaining and publishing information about

- Digital identities,
- Characteristics and presence status of agents (representing capabilities for communication and content delivery),
- Capabilities and state of entities (such as location), and
- Availability of entities for various forms of communication and the contexts in which they are available.

Establishing such a standard in the industry will facilitate creation of many inter-operable services over multiple network technologies and, in addition, allow end users greater flexibility in managing their services and communication capabilities while addressing their privacy concerns.

1.1 Motivation

Consider the following simple but desirable scenario for a communication service: An end-user wishes to receive instant messages from her management at any time on her mobile phone, from co-workers only on her desktop computer, and in certain cases for the messages to be forwarded to e-mail or even a fax machine/printer. The senders may know her availability for various forms of communication in the way she chooses to reveal it or alternatively the senders may never know how she will be receiving their messages. This scenario spans over multiple services and protocols and can only be solved currently by a proprietary solution that maintains the required information in an ad-hoc fashion within the application.

PAM is not a replacement for the protocols being standardized for various communication and network services. PAM attempts to standardize the management and sharing of presence and availability information across multiple services and networks.

The PAM specification is motivated by the observations that

- The notions of Identity, Presence and Availability are common to but independent of the various communication technologies, protocols and applications that provide services using these technologies.
- Presence does not necessarily imply availability. End-users or organizations require greater control over making themselves available through various communication devices.
- Presence based services need to address privacy concerns on who can access presence information and under what conditions.
- Management of availability will span over multiple communication services and service providers.

1.2 Goals

The main goal of Presence and Availability Management is to facilitate the development of a rich set of applications and services that span over multiple communication systems (instant messaging, e-mail, fax, telephony, etc.) and to provide the end user greater flexibility and control in managing their communications. A standardized platform allows software developers to create communication management applications that are independent of the underlying technologies and protocols.

As the next step in the evolution of directory and database enabled applications and services, separation of the management of identities and availability of users or organizations from specific applications enables uniform and centralized administration of data and creates the potential to bring control over communication services to the user’s desktops.

The purpose of this document is to publish the first release of a Presence and Availability Management interface specification created by an industry consortium, PAMforum, established for this purpose.

With a desired goal of rapid acceptance and usage, the specification has been deliberately designed to be as simple as possible with an attempt to include a minimal set of functionality that is sufficient for use in non-trivial applications. Often, this has been at the cost of some useful features, which would have made the specification baroque and cumbersome if not controversial.

1.3 Concepts

This chapter briefly describes the various concepts involved in this specification to serve as the context for the rest of the document. An UML data model of the PAM specs is provided in Appendix A.

1.3.1 Identity

Identity, for purposes of the PAM specification, is a limited electronic representation of an entity (i.e., an individual or an organization) that participates in PAM-enabled applications and services.

The main characteristic of an entity that is central to PAM specifications is the name (or handle) by which entities are identified by applications and services. Entities may have multiple names, login ids, account names, etc., by which they are identified. As PAM attempts to abstract over multiple networks and services, it does not assume that a single name will necessarily identify entities across all application domains.

Names exist in the context of a namespace. Typically, a namespace is established implicitly or explicitly by an entity (or an organization) to uniquely identify people within the domain of interest for the entity. For example, a portal provider may establish a namespace for the login names created for services within that portal. A name is assumed to be unique within a namespace. Two entities can have the same name as long as they are in different namespaces. For example, two entities may be registered under the same name in two different portal sites. A namespace is the largest set of names within which those names are uniquely assigned to identities. The PAM specification itself does not define a namespace for identities. Some PAM implementations may generate names for entities and thereby create a namespace within which they operate.

Namespaces are distinct from domains as used in some well-known naming conventions. For example, an e-mail (or SIP) handle joe@joescompany.com has joescompany.com as a domain. If entities in PAM-enabled applications and services are identified by their e-mail handles, then the entire e-mail handle is the name of a PAM identity in the global namespace of e-mail (or SIP) handles.

In some cases, the names by which people are identified are derived from their communication capabilities (e.g., e-mail address or a telephone number). Although, a device may exist with that handle, PAM distinguishes between the uses of such an address to identify an entity and the use of that address to identify an agent for communication. For example, an e-mail address can be used as an entity identifier for a travel website even if the e-mail capability is not always used for that purpose.

To enable entities to be identified by any of the names associated with them, PAM identities can be assigned aliases. A name and a namespace pair can be defined as an alias of another name and namespace pair. It is important to note that aliases are just synonyms and hence have limited semantics. In particular, they are not powerful enough to model personas each with their own capabilities and privacy requirements.

An identity can represent a single entity or a group of identities. Group identities have similar semantics to non-group identities but, in addition, maintain a list of identities that constitute the group. As an example, a sales department may be modeled as a group identity with the identities of the members of the department being member identities of the group. Group identities and their member identities do not inherit anything from each other.

No other relationships between identities are within the scope of the PAM specifications.

For flexibility and extensibility, attribute lists are used to associate additional data with identities. Identities are typed to provide a way to manage such attribute lists. An identity type may be associated with a specific set of attributes and all identities of that type inherit instances of such attributes.

PAM does not specify any pre-defined attributes or types. Applications may define and use their own identity types.

PAM implementations may map certain existing directory and database data to one or more types to allow access via PAM interfaces. PAM specifications do not specify how the data within the profiles are to be stored. They may be stored within the PAM implementation or mapped to data stored on external directories and databases.

1.3.2 Agent

An agent, for PAM purposes, is a limited electronic representation of a software or hardware device through which identities manifest themselves or make themselves available to applications and services.

An important characteristic of an agent is a list of one or more capabilities associated with it. A capability is what makes an agent useful. A capability either represents the ability of an agent to participate in communications and content delivery (e.g., instant messaging, SMS, WAP, voice) or it represents the ability of an agent to report useful information (e.g., location, velocity, temperature, mood) of the environment around it.

PAM does not specify any pre-defined capabilities. Applications may define and use their own capabilities.

Agent instances are identified by names (or handles). As for identities, names exist in the context of a namespace. Within a namespace, a name is assumed to be unique. Two agent instances can have the same name as long as they are in different namespaces. For example, a mobile phone and a PDA manufactured by two different manufacturers may coincidentally have the same serial number by which they are identified. As PAM attempts to unify services over multiple technologies, it does not assume that a name uniquely identifies agent instances across all technologies or across all manufacturers. They can be disambiguated through the use of namespaces.

No relationships between agents are within the scope of the PAM specifications.

For flexibility and extensibility, attribute lists are used to associate additional data with agents. Agents are typed to provide a way to manage such attribute lists. An agent type may be associated with a specific set of attributes and all agents of that type inherit instances of such attributes.

PAM does not specify any pre-defined attributes or types. Applications may define and use their own agent types.

PAM implementations may map certain existing directory and database data to one or more types to allow access via PAM interfaces. PAM specifications do not specify how the data within the profiles are to be stored. They may be stored within the PAM implementation or mapped to data stored on external directories and databases.

Agent instances are associated with one or more identities. This association results in the inheritance of associated agents' capabilities by the identities.

1.3.3 Presence

The concept of presence has been used in several application areas, being most explicit in Instant Messaging. Starting from a simple notion of online/offline status, it has expanded to include other context information around the status such as disposition (out to lunch, away from the computer, etc.) and activity status (on the phone, idle, etc.). Location information, on the other hand, has largely been kept separate from what has been traditionally considered presence information. PAM specifications broaden the concepts of presence recognizing that all such information, including location, describes different contexts of an entity's existence. The unifying property is that the presence information is continually changing and that there is value in knowing the current information at different points in time for services and applications.

For the purposes of PAM specifications, presence is an extensible set of characteristics that captures the dynamic context in which an identity or an agent exists at any point in time. In contrast to the relatively static information about identities or agents (e.g., names, addresses, capabilities), presence refers to dynamic information such as location, status, disposition, etc. Registrations of presence and location information in existing applications are covered by this definition.

Presence information is differentiated from the more static information associated with identities and agents that are stored in attributes. The rationalization for this design is that the presence information is dynamic and has implications on the implementation. Some of the presence information is too dynamic to be maintained in static data stores such as directories and without this hint about the data characteristics, PAM implementers may make sub-optimal decisions on the way the data is stored. Second, presence information typically has expiration data that needs to be understood by the implementation.

The PAM specification recognizes that devices that provide presence information are not necessarily devices that communicate. Certain agents may report presence information but not be capable of communication. Certain agents may be communication devices but may not be able to provide presence information. In general, the presence of an identity is computed from presence information provided by one or more agents and the ability to communicate is derived from one or more communication-capable agents available to the identity.

The PAM specification does not specify the methods by which the presence information is derived. An agent may explicitly register its own presence information or the information may be derived from other network elements. For example, an instant messaging client on a desktop computer can register its status based on when a user is logged in. A mobile phone may do an explicit registration on a WAP server for instant messaging. The phone’s presence for voice calls, on the other hand, may be inferred implicitly by querying the cellular network for the device being on when requested. The presence of an identity, on the other hand, may be computed using presence information from one or more agents.

Finally, the PAM specification does not require that the presence information be stored explicitly (i.e., in a materialized fashion) in a PAM implementation. An implementation may infer the presence information on demand from the underlying services or networks.

1.3.4 Availability

Availability is a property of an identity denoting its ability and willingness to share information about itself or to communicate with another identity based on factors such as the type of communication requested, the identity of the calling entity and the preferences and policies that are associated with the recipient. This is the primary means by which the current PAM specification enables controls for privacy. While presence is, in most applications, a necessity for availability, presence does not necessarily imply availability to all.

Availability is always with respect to a context. A context in PAM specifications is a set of attributes defining the state in which the availability is requested. For example, the query “Is Jane available for IM for Rob?” identifies the type of communication and the identity of the asker as the context. PAM allows for availability to be differentiated based on any attribute of a context. Two contexts, “Communication” and “Location” are pre-defined in PAM.

Most queries for presence in existing applications can be mapped into PAM availability queries to control the information being given out. Alternatively, queries can be mapped directly into PAM presence queries in situations where privacy controls and policies are not required or all presence data is open to the entity querying. This allows PAM specifications to be consistent with existing presence servers and to serve as the basis for presence services across multiple protocols while providing uniform and flexible privacy controls.

PAM specification does not specify whether the availability is computed on demand or stored explicitly. In some applications, the availability may be pre-computed and stored explicitly while in some, it may be computed at each request for availability.

While the PAM specification provides a mechanism to associate preferences with an Identity to control availability, it neither specifies the syntax and semantics of the preferences nor the process by which the availability is computed. These aspects are left to the implementation.

For example, a particular implementation may provide the facility to store preferences as rules such as “I prefer to receive my instant messages on my computer rather than my cell phone unless the message is from my boss or the computer is off, etc.”.

As an example, a computation of availability for communication may consist of the following algorithm:

1. Find all agents of the identity being called that are capable of the specified form of communication AND have registered their presence status as available.
2. Evaluate the rules associated with the identity being called to select the preferred agent(s) from the set of present agents determined in Step 1.
3. If there are any agents available satisfying Step 2, indicate the availability of the identity being called via the available agents.

An implementation can chose to provide one or more means to specify preferences. It is expected that if there is industry standardization on the specification of preferences, the implementations will support such a standard. This is currently outside the scope of PAM.

1.3.5 Events

Events are representations of certain identified occurrences related to the concepts described above. The PAM specification provides for registering interest (i.e., callbacks) in being notified of such occurrences. An implementation is expected to provide such notifications.

Examples of events include,

- Creation/deletion of an identity
- Association of an agent instance with an identity
- Change in presence status or location of an agent instance
- Change in availability of an identity for a particular form of communication

PAM specifications contain a set of pre-defined events. Each event is defined by a name of the event, a set of input attribute value pairs that must be provided when an event is registered for and a set of attribute value pairs that are included in the notifications sent out when the event of interest occurs.

1.4 Scope

Presence and Availability Management has the following types of information in its scope:

- Identities, which consist of names and aliases of entities participating in communications.
- Agent information, which consists of names and communication capabilities of software and/or hardware devices.
- Agent provisioning, which consists of associations between instances of agents and identities.
- Presence information, which consists of an identity's or an agent's dynamic characteristics such as status and geographical location.
- Availability information, which consists of preferences associated with identities and computation of availability, based on the devices present and the current preferences.
- Notification of changes to the above pieces of information.
- Security issues for access to this information.

The PAM specification consists of interfaces to manage or access the above information.

The specification purposefully does not include

- Storage design or storage requirements for any of the presence and availability information.
- Protocol specification to access the interfaces.

These are to be decided by specific implementations of the PAM specification.

2 Model

This Chapter describes the model for Presence and Availability Management that has influenced the design of the specifications. The model embodies assumptions about the architecture in typical usage, the security and privacy issues, the types of clients or applications that will access PAM implementations, and the framework in which they do so.

Presence and Availability Management has dual roles. In one role, it acts as an abstraction layer (Fig. 1) that sits between

- The end-users who will manage their communication identities and availability,
- The communication services that will behave according to the wishes of the end-user, and
- The communication networks to which the end-user's devices are attached and from which their status is to be obtained or inferred.

Here the goal is to "keep the end-user in the loop" i.e., to let the end-users manage their communication services as much as the service providers manage their subscribers. This role primarily determines the functionality of the specifications.

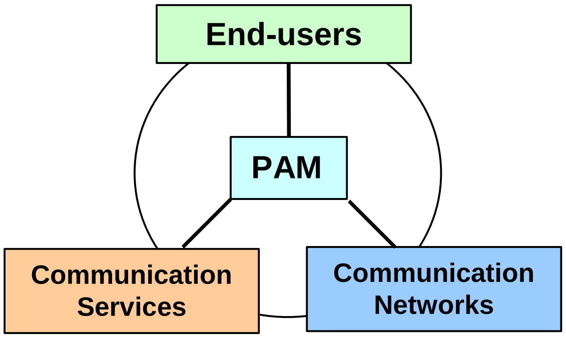


Fig.1 "Keeping end-users in the loop"

In the second role, Presence and Availability Management provides the means for multiple administrative domains to share information about identity, presence and availability in controlled ways (Fig. 2). This sharing may occur for the purposes of allowing communications between end-users in multiple domains and/or for the purposes of allowing the information to be federated into a merged global address space. In either case, this role determines the security and privacy aspects of the specification design.

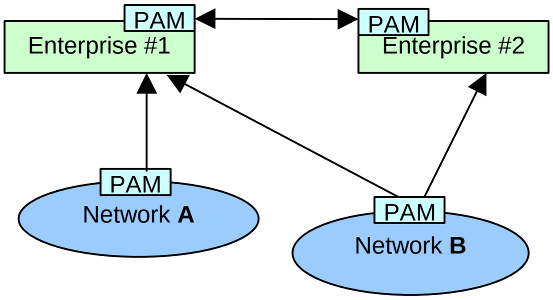


Fig.2 Information sharing via PAM interfaces

2.1 Architecture

There are several architectural scenarios for use of PAM depending on the participants being legacy systems or future systems or a combination of both. The least intrusive use of PAM is through an implementation of an abstraction layer over existing legacy systems to allow some limited management control for the end-user (Fig. 3). The limited benefits come from being able to use third-party end-user management software written for the standard specification. The legacy systems themselves are unaware of the PAM layer. Example applications include number translation schemes in telephony systems enhanced through PAM interfaces, LDAP enabled e-mail systems extended through PAM layer to provide dynamic and/or customized address information, etc.

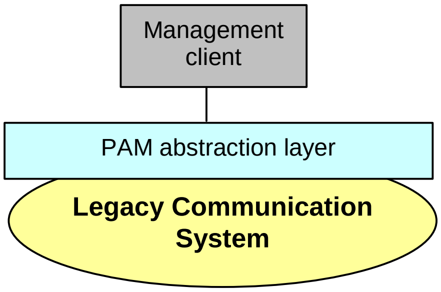


Fig.3 Abstraction over legacy systems

The next level of adoption comes from communication services or networks that already have some notion of identity, presence and/or availability, exporting this information through the PAM interface (Fig. 4). This allows third-party end-user management software written for the standard specification to manage across multiple communication systems on behalf of the user. As the communication services are already designed to take some of these notions into account, the end-user is able to customize the services to a larger extent than the previous scenario. Example applications include Instant Messaging, Email, etc.

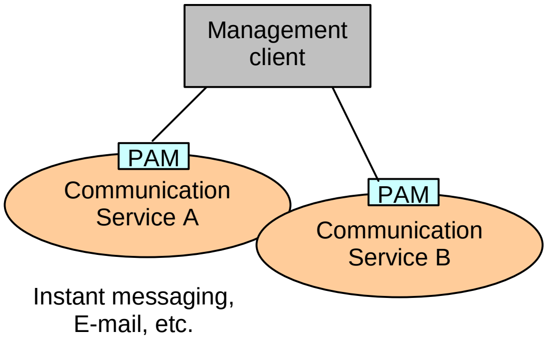


Fig.4 Exporting from communication services

Maximum benefits from Presence and Availability Management comes in a scenario (Fig. 5) in which

- Communication networks export relevant information and device status through PAM interface,
- Communication services are written to consult PAM servers to affect communication handling, and
- End-user management systems written to PAM specifications allow end-users to specify policies and preferences for their communication capabilities.

Any of the communication services (voice telephony, fax, e-mail, instant messaging, etc.) can potentially use this architecture.

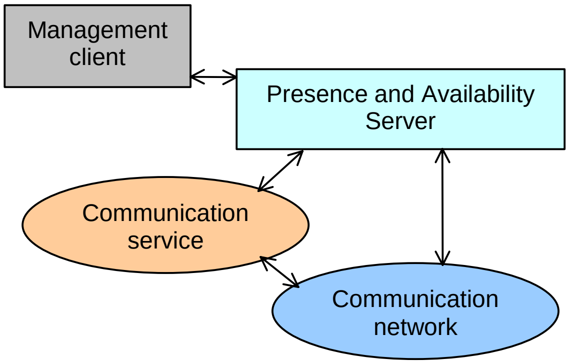


Fig.5 PAM enabled communications

2.2 Security and privacy

As the Presence and Availability Management interface is designed to share information across administrative domains and to facilitate availability computation based on the identity of the entity desiring communication, security and privacy issues are addressed in the design. The issues considered to be within the scope of PAM are:

- Access control to an implementation of the PAM specification.
- Use of an authenticated entity's credentials by methods in the specification.
- Mandated fields in information supplied by a PAM implementation to describe the expected degree of privacy under which the information is provided.

To understand the distinction between the first two issues, consider, for example, an end-user that logs on to an Instant Messaging client and wishes to send a message. The client (or a gateway to which the client talks to) may access a PAM implementation to determine the availability of the destination for the message. The client (or the gateway) will need to be authorized for access to the PAM implementation independent of the user that logs in. A gateway may, in fact, do this access on behalf of a number of clients and, for performance reasons, wish to authenticate itself just once on start up rather than at each invocation. Second, each invocation of a particular method to check for the availability will need to contain the credentials of the end-user that logged into the client so that the computation of the availability can take that into account when necessary for privacy issues.

It should be noted that the PAM specification allows for the possibility that the authentication of the end-users is not necessarily done within the PAM implementation itself. As long as the authenticated credentials supplied by the client (or gateway) are acceptable for validation and the client (or the gateway) itself is authenticated by the implementation, the authentication of end-users can occur anywhere outside the PAM implementation. A deployment scenario for a particular application is that one or more authentication services are provided as external services over PAM implementations.

This design does not preclude the possibility that the client (or the gateway) cannot be authenticated. Therefore, the credentials supplied by the client (or the gateway) may be held to stronger authentication criterion than credentials supplied by a trusted client (or gateway).

Finally, the PAM specification does not mandate the use of authentication within an implementation if the environment in which it is used does not require it.

Privacy issues are addressed primarily by providing a mechanism to control the information flowing out of a PAM implementation based on whatever criterion the end user may choose to specify in the availability preferences and independent of any particular application.

Once the information flows out a PAM implementation, its distribution is outside the control of the end user. However, to specify an expectation of the degree of privacy under which the information is provided, PAM mandates a field within the information profiles provided. This field can state privacy conditions such as "Not for further distribution", "One time use only", etc. The enforcement of these restrictions is beyond the scope of PAM.

The following security issues were considered to be outside the scope of PAM:

- Authentication of the identity of the end-users or entities. As explained above, this authentication may be provided by a third-party authentication service or it may occur through an authentication service written over the PAM platform. The only requirement is that the type of credentials supplied by the authentication service be acceptable to the PAM platform implementation being accessed.
- Encryption of the flow of information between a PAM platform implementation and clients of this implementation. This is dependent on the method of access to the interface which is outside the scope of the PAM specification and hence to be determined by the implementation.

2.3 Access framework

The purpose of the access framework is to provide the interface for the features independent of the Presence and Availability Management functions but necessary for the use of a PAM platform implementation. These features cover

- Initial contact
- Authentication
- Access to the PAM interfaces
- Discovery methods

Rather than “re-invent the wheel”, the design of the access framework has been fashioned after the Parlay [www.parlay.org] framework design but simplified considerably to be commensurate with the simpler scope of these specifications. In particular, we assume that the interfaces defined in these specifications are the only services provided by the platform and services written on top of the PAM platform have their own detection, authentication and access mechanisms. A second departure from the Parlay framework is that the authentication is optional and an implementation may provide access to its interfaces even without any authentication if it wishes to do so.

In addition, depending on the access protocols used in a particular implementation, the platform implementation may provide direct access to the interfaces (perhaps as messages to a port) without requiring the initial contact or authentication. As we will see in the next section, a simple client application that wishes to check for availability of an end-user before attempting communication should not have to necessarily go through the initial contact and authentication steps (akin to an implicit bind in LDAP queries). The framework exists to direct those implementations that wish to expose the interfaces under much tighter control. An implementation can also provide a more restricted interface for clients with no authentication compared to those that authenticate themselves.

The PAM specification does not specify the authentication mechanism to be used as this is left to the implementation, which may support one or more authentication mechanisms.

The discovery methods allow the applications to discover the implementation-dependent features and capabilities of the PAM implementation such as interfaces supported, authentication methods supported, events supported, etc.

2.4 Levels of access

As described in earlier sections, the Presence and Availability Management platform can be used for a variety of purposes including

- Platform for third-party communication management software for end-users.
- Single-point privacy administration for enterprises.
- Federating namespaces across multiple communication services and networks.
- Exporting enterprise-managed identity, presence and/or availability data for use by external communication services/devices.
- Exporting status and/or location data of devices from networks for use by communication services.

Not all methods in every interface are likely to be used in every context. For example, a simple communication device such as an instant messaging client can check for availability of an end-user using the availability interface but may not need the preference management methods. Communication management software for the end-user may manage the preferences for the user but would not require the user identity creation methods. Enterprise administration software or services installed on top of the platform may require access to all the methods in every interface.

While the interfaces defined in this specification are a minimal union of all the potential types of accesses, it is useful to recognize three categories of platform access primarily differentiated by the authentication and security requirements as described below. Unlike Parlay specifications, PAM specification does not partition the methods and interfaces into views for each type of access. Platform implementations that implement the access framework or define the access protocols are encouraged to define views for the three types of accesses and decide on the subset of methods supported in each view. PAM specification does not require that every method be supported in implementations. Consequently, software written for the platform must take into account the possibility that any method may return with a “not supported” status.

Briefly described below are the characteristics of the three major levels of accesses:

2.4.1 Application

Applications are independent, stand-alone software systems that access PAM implementations on a continuous basis possibly on behalf of multiple clients. Examples include PAM-compliant gateways, switches, messaging servers, address translation systems, enterprise management systems, etc. These applications are likely to be “always-on” and must be allowed to authenticate themselves only at startup of the applications. The applications are typically expected to be non-trivial and running on computing platforms that allow for heavyweight frameworks such as CORBA to be used for access to the interfaces.

2.4.2 Service

Services are software modules that extend or provide additional functionality to PAM implementations. They may run in the same process space as the PAM platform or may run remotely. It is up to the implementations to provide implementation-specific mechanisms for adding, using or managing such services. Examples of services include “buddy list” systems, identity authentication systems, etc.

2.4.3 Thin client

Independent software differentiated from applications in being relatively lightweight and potentially running on devices with minimal computing capabilities such as cellular phones or palm devices. The use of the PAM implementation by a thin client is expected to be infrequent and transitory. Consequently, authentication, when required, may need to happen as often as an individual method access. The PAM implementations need to provide lightweight, preferably message based access for such clients. The thin client access may not be limited to checks for presence and/or availability alone. They may provide some limited end-user preference management capabilities as well.

2.5 Use cases

This section gives some illustrative use cases of the different interfaces in this document. This list is not meant to be an exhaustive listing of all the possible use cases but rather an aid in understanding possible uses of the PAM interfaces.

2.5.1 Identity Management

1. Create an identity for John Smith (identity name) of type subscriber (identity type) with the following address information (a specific named profile for addresses).
2. Delete the identity for John Smith

3. Create an alias of Jsmith@company.com under the namespace of "myISP" (namespace created by the ISP with whom John has an acoount)
4. Delete the alias of Jsmith@company.com for John Smith
5. Create a group named “Project Manhattan” (group identity name) of type projects (identity type) with the following people in it – John, ... (group members)
6. Delete the group called “Project Manhattan”
7. Add John Smith to “Project Manhattan” Group
8. Remove John Smith from “Project Manhattan” Group
9. Is there a John Smith in the default namespace?
10. What is the address information for John Smith?
11. What are John Smith's aliases?
12. What groups does John Smith belong to?
13. Whose alias is Jsmith@myISP.com under the name space of "myISP"?
14. Who are the members of “Project Manhattan”?

2.5.2 Agent Management

1. Create an agent of type “mobile phone” with the ID 123456789 (agent name) and the following capabilities - WAP, Voice, SMS
2. Delete the agent with the ID 123456789
3. Add the capability "video conferencing" to the agent with id 123456789
4. Disable the capability "WAP" for the agent with id 123456789
5. Associate the agent with ID 123456789 with the agent type "GSM phone".
6. Is there an agent with ID 123456789?
7. What agent types are associated with the agent ID 123456789?
8. What is Agent with ID 123456789 capable of?
9. Is Agent with ID 123456789 capable of SMS?

2.5.3 Agent Assignment

1. John Smith now has the phone with ID 123456789
2. John Smith no longer has the phone with ID 123456789
3. What agents does John Smith have for instant messaging (capability)?
4. Is John Smith able to receive instant messages (capability)?
5. What all means can John Smith be communicated with?
6. Who is currently assigned to the phone with ID 123456789?

2.5.4 Agent Presence

1. Set agent status information (dynamic attributes) for mobile phone 123456789 for Instant Messaging with the following location information valid until removed.
2. Set the expiration time for the location information (dynamic attribute) for mobile phone 123456789
3. Update status information (a specific presence profile) of mobile phone 123456789 for Instant Messaging.

4. What is the call status of phone 123456789 for voice calls?
5. What is the location information about phone 123456789 for SMS?
6. What is the motion information of phone 123456789 for SMS?

2.5.5 Identity Presence

1. Set status information for John Smith for Instant Messaging valid until removed.
2. Set call status information of John Smith for voice calls.
3. Set location information of John Smith for Instant Messaging.
4. What is the call status of John Smith for voice calls?
5. What is the location information about John Smith for SMS?
6. What is the motion information of John Smith for SMS?

2.5.6 Availability

1. Store these preferences for John Smith for his voice calls.
2. Delete John Smith's preferences for voice calls.
3. What are John Smith's preferences for voice calls?
4. Is John Smith available to talk to Jane Doe on the phone?
5. How can Jane Doe contact John Smith via Instant Messaging?
6. What is John Smith's location information for Jane Doe?

3 PAM Interfaces

3.1 Identity Management

This section describes the programmatic interface to Identity Management. 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.

3.1.1 Summary

Identity methods:

PAM_IM_createIdentity	create a new identity
PAM_IM_deleteIdentity	delete an existing identity
PAM_IM_isIdentity	check if an identity exists with the specified name

Group identity methods:

PAM_IM_createGroupIdentity	create a new group identity
PAM_IM_deleteGroupIdentity	delete an existing group identity
PAM_IM_addToGroup	add an identity to a group identity
PAM_IM_removeFromGroup	remove an identity from a group identity
PAM_IM_listMembers	list member identities of group
PAM_IM_isGroupIdentity	check if an identity is a group identity
PAM_IM_listGroupMembership	list groups an identity is member of

Alias methods:

PAM_IM_addAlias	add an alias to an identity
PAM_IM_removeAlias	remove an alias for an identity
PAM_IM_listAliases	list all aliases of an identity
PAM_IM_lookupByAlias	lookup identities that have the specified alias

Type association and attribute methods

PAM_IM_associateTypes	associate an identity instance with the specified types
PAM_IM_disassociateTypes	remove the association of a type with an identity instance.
PAM_IM_listTypesOfIdentity	list the types associated with an identity
PAM_IM_hasType	check if the identity has the specified type

PAM_IM_getIdentityAttributes	retrieve attributes associated with the identity
PAM_IM_setIdentityAttributes	modify the attributes associated with the identity.

PAM_IM_addAlias

Add an alias in the specified namespace to an existing Identity.

Inputs

- ***identity*** **PAM_T_FQName** specifies the Identity to which the alias will be added.
- ***alias*** **PAM_T_FQName** specifies the alias to be added.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_ALIAS_EXISTS** indicates that the specified alias is already associated to the Identity.
- **PAM_ALIAS_NOT_UNIQUE** indicates that the alias has already been assigned to another identity
- **PAM_UNKNOWN_IDENTITY** indicates that specified Identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The alias domain name must be specified in Alias parameter.

The identity can be a group identity.

PAM_IM_addToGroup

Add an existing identity to a group identity.

Inputs:

- ***group*** **PAM_T_FQName** specifies the group Identity to which the member will be added.
- ***member*** **PAM_T_FQName** specifies the identity to be added as a member of the group.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_MEMBER_EXISTS** indicates that the specified member is already in the group.
- **PAM_UNKNOWN_GROUP** indicates that the specified group identity does not exist.
- **PAM_UNKNOWN_MEMBER** indicates that the specified member identity does not exist.
- **PAM_IS_CYCLIC** indicates that the requested operation will create cyclic relationship
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_IM_associateTypes

Associate an identity instance with the specified types.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the name of the identity
- ***identityTypes*** **PAM_T_String[]** specifies the names of the type to be associated.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified Identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that a specified type has not been defined.
- **PAM_TYPE_ASSOCIATED** indicates that a named type has already been associated with the identity.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_IM_createGroupIdentity

Create a new Group Identity with the specified name.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the group Identity to be created.
- ***identityTypes*** **PAM_T_String[]** specifies the group's associated types. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_IDENTITY_EXISTS** indicates that the specified Identity already exists.
- **PAM_UNKNOWN_TYPE** indicates that one of the specified types has not been defined.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

Name must be unique across both group identities and non-group identities.

Names must be unique across the same types.

PAM_IM_createIdentity

Create a new non-group Identity with the specified name.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be created.
- ***identityTypes*** **PAM_T_String[]** specifies the identity's associated types. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_IDENTITY_EXISTS** indicates that the specified Identity already exists.
- **PAM_UNKNOWN_TYPE** indicates that one of the specified types has not been defined.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

Names must be unique across both group identities and non-group identities.

Names must be unique across types within a namespace.

PAM_IM_deleteGroupIdentity

Delete the specified group identity and all its related data.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the group identity to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified group identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_IM_deleteIdentity

Delete the specified identity and all its related data.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified non-group identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_IM_disassociateTypes

Remove the association of a type with an identity instance.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity
- ***identityTypes*** **PAM_T_String[]** specifies the names of the types to be removed.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.
- **PAM_DISASSOCIATED_TYPE** indicates that one of the specified types is not associated with the named identity.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The definition of the type itself remains unaffected and the types may continue to be associated with other identities.

PAM_IM_getIdentityAttributes

Return the attributes associated with the identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***identityType*** **PAM_T_String** specifies the type of the identity with which the required attributes are associated. Is optional.
- ***attributeNames*** **PAM_T_String[]** list of attributes of interest. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_Attribute[]** 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.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named identity has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes is not part of the specified type.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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

PAM_IM_hasType

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be checked.
- ***typeName*** **PAM_T_String** specfies the type to be checked for
- ***authToken*** **PAM_T_Credential** of the caller who is making the request.

Output:

- ***hasType*** **PAM_T_Boolean** true if an identity with the specified name has the named type associated with it, false otherwise.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_isGroupIdentity

Check if the specified group identity exists.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be checked.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***isIdentity*** **PAM_T_Boolean** true if a group identity with the specified name exists, false otherwise.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The method returns false for non-group identities.

PAM_IM_isIdentity

Check if the specified non-group identity exists.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be checked.
- ***authToken*** **PAM_T_Credential** of the caller who is making the request.

Output:

- ***isIdentity*** **PAM_T_Boolean** true if an identity with the specified name exists and false otherwise.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The method returns false for group identities.

PAM_IM_listAliases

List the aliases for the specified Identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be looked up.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

- ***aliases*** **PAM_T_FQName[]** is an array containing all aliases to the specified Identity. An array with zero elements is returned if there are no aliases associated with the identity.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_listGroupMembership

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be looked up.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

- ***groupIdentities*** **PAM_T_FQName[]** is an array containing all groups the specified Identity is a member of. An array with zero elements is returned if the specified identity is not a member of any group.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_listMembers

List the members of the specified group Identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the group Identity whose members are required.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

- ***members*** **PAM_T_FQName[]** is an array containing all members of the specified group Identity. An array with zero elements is returned is there are no members.

Return Status:

- **PAM_UNKNOWN_GROUP** indicates that the specified group identity does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_listTypesOfIdentity

List the types associated with the specified Identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the Identity to be looked up.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

- ***typeNames*** **PAM_T_String[]** is an array containing all types associated with the specified Identity. An array with zero elements is returned if there are no types associated with the identity.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_lookupByAlias

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

Inputs:

- ***alias*** **PAM_T_FQName** specifies the alias to be looked up.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

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

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IM_removeAlias

Remove the specified alias from an existing Identity.

Inputs:

- ***identity***	**PAM_T_FQName** specifies the Identity from which the alias will be deleted.
- ***alias***	**PAM_T_FQName** specifies the alias to be deleted.
- ***authToken***	**PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.
- **PAM_UNASSIGNED_ALIAS** indicates that the specified alias was not an alias of the named identity.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the alias is not uniquely assigned, this method does not affect other identities which may have the same alias.

PAM_IM_removeFromGroup

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

Inputs:

- ***group*** **PAM_T_FQName** specifies the group Identity from which the member will be removed.
- ***identity*** **PAM_T_FQName** specifies the identity to be removed as a member of the group.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_NOT_MEMBER** indicates that the specified member is not a member of the group.
- **PAM_UNKNOWN_GROUP** indicates that the specified group does not exist.
- **PAM_UNKNOWN_MEMBER** indicates that the specified member identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

Upon successful completion, the specified group identity will not contain the member identity in its group.

PAM_IM_setIdentityAttributes

Modify the attributes associated with the named Identity.

Inputs:

identity	**PAM_T_FQName** specifies the Identity.
identityType	**PAM_T_String** specifies the type of the identity for the operation. Is optional.
attributes	**PAM_T_Attribute[]** contains the list of attributes and their values.
authToken	**PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named identity has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTES** indicates that the specified attribute list contains attributes not part of the named type.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

3.2 Agent Management

This section describes the programmatic interface to Agent Management. 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.

3.2.1 Summary

Agent methods:

PAM_AM_createAgent	create a new agent
PAM_AM_deleteAgent	delete an existing agent
PAM_AM_isAgent	check if an agent exists with the specified name

Agent capability methods:

PAM_AM_enableCapabilities	enable specified capabilities of agent
PAM_AM_disableCapabilities	disable specified capabilities of agent
PAM_AM_listEnabledCapabilities	list enabled capabilities of agent
PAM_AM_listAllCapabilities	list all capabilities of agent
PAM_AM_isCapableOf	check if agent has specified capabilities

Type association and attribute methods

PAM_AM_associateTypes	associate an agent instance with the specified types
PAM_AM_disassociateTypes	remove the association of a type with an agent instance.
PAM_AM_listTypesOfAgent	list the types associated with an agent
PAM_AM_hasType	check if the agent has the specified type
PAM_AM_getAgentAttributes	retrieve attributes associated with the agent
PAM_AM_setAgentAttributes	modify the attributes associated with the agent.

PAM_AM_associateTypes

Associate an agent instance with the specified types.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the name of the agent
- ***agentTypes*** **PAM_T_String[]** specifies the names of the type to be associated.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified agent does not exist.
- **PAM_UNKNOWN_TYPE** indicates that a specified type has not been defined.
- **PAM_TYPE_ASSOCIATED** indicates that a named type has already been associated with the agent.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The agent 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.

PAM_AM_createAgent

Create an agent initialized with the specified capabilities.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the name of Agent to be created.
- ***agentTypes*** **PAM_T_String[]** specifies the types of the Agent to be created. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_AGENT_EXISTS** indicates that an Agent with the *agentName* already exists.
- **PAM_UNKNOWN_TYPE** indicates that a specified type name has not been defined as an agent type.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_deleteAgent

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

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the Agent with the specified name does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_disableCapabilities

Disable the specified capability of the Agent.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent.
- ***capabilities*** **PAM_T_Capability[]** specifies the capability to be disabled.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Outputs:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the Agent with the specified identifier does not exist.
- **PAM_NO_CAPABILITY** indicates that the specified agent does not have the capability to be disabled.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_disassociateTypes

Remove the association of a type with an agent instance.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent
- ***agentTypes*** **PAM_T_String[]** specifies the names of the types to be removed.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the specified agent does not exist.
- **PAM_DISASSOCIATED_TYPE** indicates that one of the specified types is not associated with the named agent.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The definition of the type itself remains unaffected and the types may continue to be associated with other agents.

PAM_AM_enableCapabilities

Enable the specified capabilities of the agent.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the name of the Agent.
- ***capabilities*** **PAM_T_Capability[]** specifies the capabilities to be enabled.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Outputs:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the Agent with the specified name does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_getAgentAttributes

Return the attributes associated with the agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***agentType*** **PAM_T_String** specifies the type of interest. Is optional.
- ***attributeNames*** **PAM_T_String[]** list of attributes of interest. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_Attribute[]** contains the list of specified attributes and their values. If the attributeNames parameter is an empty array, all attributes in the named agent are output.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named agent has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes is not part of the specified type.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the type is not specified, all associated types are assumed.

PAM_AM_isAgent

Check if the specified agent exists.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent to be checked.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***isAgent*** **PAM_T_Boolean** true if an agent with the specified name exists, false otherwise.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_isCapableOf

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

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent to be checked.
- ***capability*** **PAM_T_Capability** capability to be checked.
- ***authToken*** **PAM_T_Credential** of the caller who is making the request.

Output:

- ***isCapable*** **PAM_T_Boolean** true if the agent has the specified capability, false otherwise.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified agent does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_hasType

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

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent to be checked.
- ***typeName*** **PAM_T_String** specfies the type to be checked for
- ***authToken*** **PAM_T_Credential** of the caller who is making the request.

Output:

- ***hasType*** **PAM_T_Boolean** true if an agent with the specified name has the named type associated with it.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the specified agent does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_listAllCapabilities

List the capabilities for the specified Agent.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent whose capabilities are to be listed.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***capabilities*** **PAM_T_Capability[]** is the list of capabilities for the Agent. Returns a zero-length array if no capabilities exist for the agent.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified agent does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

Both enabled and disabled capabilities are returned.

PAM_AM_listEnabledCapabilities

List the enabled capabilities for the specified Agent.

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the Agent whose capabilities are to be listed.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***capabilities*** **PAM_T_Capability[]** is the list of enabled capabilities for the Agent. Returns a zero-length array if no enabled capabilities exist for the agent.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified agent does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_listTypesOfAgent

List the types associated with the specified agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent to be looked up.
- ***authToken*** **PAM_T_Credential** of the entity who is making the request.

Output:

- ***typeNames*** **PAM_T_String[]** is an array containing all types associated with the specified agent. An array with zero elements is returned if there are no types associated with the agent.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the specified agent does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AM_setAgentAttributes

Modify the attributes associated with the named agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***agentType*** **PAM_T_String** specifies the type of the agent for the operation. Is optional.
- ***attributes*** **PAM_T_Attribute[]** contains the list of attributes and their values.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that specified agent does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named agent has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTES** indicates that the specified attribute list contains attributes not part of the named type.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

3.3 Agent Assignment

This Section describes the programmatic interface to Agent Assignment. 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.

3.3.1 Summary

Assignment methods:

PAM_AA_assignAgent	assign an agent to an identity
PAM_AA_unassignAgent	unassign agent from an identity
PAM_AA_listAssignedAgent	list agents assigned to an identity
PAM_AA_listAssociatedIdentitiesOfAgent	list all identities that have the specified agent associated with them

Inherited capability search methods:

PAM_AA_listAssignedAgentsByCapability	list associated agents with specified capability
PAM_AA_listCapabilitiesOfIdentity	list capabilities of an identity
PAM_AA_isIdentityCapableOf	check if an identity has the specified capability

PAM_AA_assignAgent

Assign an existing agent to an existing identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity to assign the agent to.
- ***agentName*** **PAM_T_FQName** specifies the Agent.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the specified identity does not exist.
- **PAM_UNKNOWN_AGENT** indicates that the specified agent does not exist.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_isIdentityCapableOf

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity of interest.
- ***capability*** **PAM_T_Capability** identifies the capability to check for.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***isCapable*** **PAM_T_Boolean** returns true if the identity has this capability, false otherwise.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_listAssignedAgents

List the Agents assigned to an identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***agentNames*** **PAM_T_FQName[]** the list of agent names assigned to the identity. An array with zero elements is returned if no agents are assigned to the identity.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_listAssignedAgentsByCapability

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***capability*** **PAM_T_Capability** is the capability of interest.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***agentNames*** **PAM_T_FQName[]** the list of agent names with the specified capability. An array of zero elements is returned no agents are found.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_listAssociatedIdentitiesOfAgents

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

Inputs:

- ***agentName*** **PAM_T_FQName** specifies the agent.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***identityNames*** **PAM_T_FQName[]** the list of identities that have been assigned the specified agent. An array with zero elements is returned if no identities have been assigned this agent.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_listCapabilitiesOfIdentity

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity of interest.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***capabilities*** **PAM_T_Capability[]** the list of the identity's capabilities. Returns an array of zero elements if no capabilities exist.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AA_unassignAgent

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

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity of interest.
- ***agentName*** **PAM_T_FQName** specifies the agent.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_ UNKNOWN_ASSIGNMENT** a warning indicating that no assignment exists for this identity and agent.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

3.4 Agent Presence

This Section describes the programmatic interface to Agent Presence. The purpose of this interface is to maintain the dynamic presence information of agents.

The underlying implementations may optimize the storage for this dynamic data rather than rely on a general-purpose directory or database when performance is an issue. Agents may explicitly register the presence information or the presence information may be implicitly derived from the underlying networks.

The presence information is modeled through dynamic attributes. Sets of dynamic attributes can be defined per agent type (e.g., agent location, power status) or per agent capability (e.g., agent status for voice/messaging, communication address).

This interface is meant for use by applications that query and update agent presence information directly regardless of the identities to which the agent is assigned.

3.4.1 Summary

Agent presence methods:

PAM_AP_setAgentPresence	Set presence attribute values for an agent.
PAM_AP_setCapabilityPresence	Set presence attribute values for a set of capabilities of an agent.
PAM_AP_setAgentPresenceExpiration	Set or reset the expiration of named presence attributes for an agent.
PAM_AP_setCapabilityPresenceExpiration	Set or reset the expiration of named presence attributes for a set of capabilities of an agent.
PAM_AP_getAgentPresence	Retrieve named presence attributes for an agent.
PAM_AP_getCapabilityPresence	Retrieve named presence attributes for a capability of an agent.

PAM_AP_setAgentPresence

Set presence attribute values for an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***agentType*** **PAM_T_String** specifies the type of the agent.
- ***attributes*** **PAM_T_Attribute[]** specifies the dynamic attributes to set.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named agent has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a supplied attribute is not a dynamic attribute of the specified agent. May be returned if either the name and/or type of a supplied attribute does not match any dynamic attribute of the specified agent. No attributes are affected.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AP_setCapabilityPresence

Set presence attribute values for a set of capabilities of an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***capability*** **PAM_T_Capability** specifies which capability of the agent to set.
- ***attributes*** **PAM_T_Attribute[]** specifies the dynamic attributes to set.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_UNKNOWN_CAPABILITY** indicates that a supplied capability is not a capability of the requested agent. No attributes are affected.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a supplied attribute is not a dynamic attribute of the specified capability. May be returned if either the name and/or type of a supplied attribute do not match any dynamic attribute of the specified capability. No attributes are affected.
- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AP_setAgentPresenceExpiration

Set or reset the expiration of named presence attributes for an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***agentType*** **PAM_T_String** specifies the type of the agent.
- ***attributeNames*** **PAM_T_String[]** specifies the names of the dynamic attributes. May be an empty array to indicate all dynamic attributes are to be affected.
- ***expiresIn*** **PAM_T_TimeInterval** specifies the number of seconds until the attributes expire. A value of –1 indicates no expiration. A value of 0 indicates immediate expiration.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named agent has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a supplied attribute name is not a dynamic attribute of the specified agent. No attributes are affected.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AP_setCapabilityPresenceExpiration

Set or reset the expiration of named presence attributes for a set of capabilities of an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***capability*** **PAM_T_Capability** specifies which capability of the agent to affect.
- ***attributeNames*** **PAM_T_String[]** specifies the names of the dynamic attributes. May be an empty array to indicate all dynamic attributes are to be affected.
- ***expiresIn*** **PAM_T_TimeInterval** specifies the number of seconds until the attributes expire. A value of –1 indicates no expiration. A value of 0 indicates immediate expiration.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_NO_CAPABILITY** indicates that a supplied capability is not a capability of the requested agent. No attributes are affected.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a supplied attribute name is not a dynamic attribute of the specified agent. No attributes are affected.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AP_getAgentPresence

Retrieve named presence attributes for an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***agentType*** **PAM_T_String** specifies the type of the agent.
- ***attributeNames*** **PAM_T_String[]** specifies the dynamic attributes of interest. Can be an empty array to indicate all dynamic attributes are to be retrieved.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_Attribute[]** contains the requested dynamic attributes associated with the specified agent. If the attributeNames parameter is an empty array, all dynamic attributes of the specified agent are included.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named agent has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that one of the named attributes is not associated with the specified agent.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_AP_getCapabilityPresence

Retrieve named presence attributes for a capability of an agent.

Inputs:

- ***agent*** **PAM_T_FQName** specifies the agent.
- ***capability*** **PAM_T_Capability** specifies which capability of the agent for which attributes are desired.
- ***attributeNames*** **PAM_T_String[]** specifies the dynamic attributes of interest. Can be an empty array to indicate all dynamic attributes are to be retrieved.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_Attribute[]** contains the requested dynamic attributes associated with the specified agent. If the attributeNames parameter is an empty array, all dynamic attributes of the specified agent are included.

Return Status:

- **PAM_UNKNOWN_AGENT** indicates that the requested agent does not exist.
- **PAM_UNKNOWN_CAPABILITY** indicates the requested agent does not have the requested capability.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that one of the named attributes is not associated with the specified agent.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

3.5 Identity Presence

This Section describes the programmatic interface to Identity Presence. The purpose of this interface is to maintain the dynamic presence information of identity.

The underlying implementations may optimize the storage for this dynamic data rather than rely on a general-purpose directory or database when performance is an issue. Presence information for identities may be explicitly registered are may be implicitly derived from the underlying networks or presence information from agents associated with the identity.

This interface is meant for use by applications that register and/or maintain dynamic presence information associated with identities and accessible without the privacy or other controls established by availability preferences. These applications may not be aware of the name and the types of agents associated with the identity.

The presence information can be explicitly registered using the interface or the presence may come from information implicitly derived (e.g., using presence information of agents associated with the identity).

3.5.1 Summary

Identity presence methods:

PAM_IP_setIdentityPresence	Set presence attribute values for an identity.
PAM_IP_setIdentityPresenceExpiration	Set or reset the expiration of named presence attributes for an identity.
PAM_IP_getIdentityPresence	Retrieve named presence attributes for an identity.

PAM_IP_getIdentityPresence

Retrieve presence attributes associated with an identity.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***identityType*** **PAM_T_String** specifies the identity type.
- ***attributeNames*** **PAM_T_String[]** specifies the attributes of interest. Can be an empty array
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_Attribute[]** contains the requested attributes of the named capability. If the attributes parameter is an empty array, all attributes of the named profile are included.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the requested identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named identity has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that one of the named attributes is not associated with the named capability.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_IP_setIdentityPresence

Set identity's dynamic attributes.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***identityType*** **PAM_T_String** specifies the type of the identity.
- ***attributes*** **PAM_T_Attribute[]** specifies the attributes to set.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the requested identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named identity has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a supplied attribute is not associated with the named capability.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

PAM_IP_setIdentityPresenceExpiration

Set or reset the expiration of an identity's named presence attributes.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity.
- ***identityType*** **PAM_T_String** specifies the type of the identity.
- ***attributeNames*** **PAM_T_String[]** specifies the names of the attributes. Can be an empty array.
- ***expiresIn*** **PAM_T_TimeInterval** specifies the number of seconds until the attributes expire. A value of –1 indicates no expiration.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that the requested identity does not exist.
- **PAM_UNKNOWN_TYPE** indicates that the named identity has not been associated with the named type.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that a named attribute is not associated with the named capability.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the attributeNames parameter is an empty array, the expiration time of all attributes defined for the capability will have their expiration time changed.

3.6 Availability

This Section describes the programmatic interface to Availability Management. The purpose of the interface is to

- Manage the preferences specified for the availability of an identity and, to
- Query for the availability of identities for specific capabilities.
- Query for information about identities.

Simple implementations may equate the availability of identities to presence of their agents with available status. More complex implementations may consider, in addition, the preferences specified for availability as well as the attributes of the entity asking for availability.

The queries for availability are done for a specified *context*. A context is a set of attributes describing the situation for which availability is requested. PAM specifies two pre-defined contexts – *Communication* and *Location*. The latter is used for the availability of the location information. The former is used for availability for a specific mode of communication. Applications and PAM implementations may extend and provide additional contexts such as availability at a particular location, availability for a specific mode of communication at a given location, etc. The context information also includes any information about the asker as may be provided by the asker.

The specification does not define the type and format of preferences. Implementations will decide what type of preference engines/computations to use and hence specify the types of preferences they can handle. Some, for example, may allow rules in some specific language, while some may allow for program fragments with certain interfaces to be deposited as individual preference computations.

In other words, the algorithm for availability computations is entirely up to the implementation and is transparent to the entities checking for availability.

3.6.1 Summary

Availability Query methods:

PAM_AV_getAvailability	Get the availability of an identity

Preference Management methods:

PAM_AV_getPreference	Get preferences associated with an identity
PAM_AV_setPreference	Set preferences for an identity

PAM_AV_getAvailability

Get the availability for an identity for a given context.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity for which the availability is being requested.
- ***context*** **PAM_T_Context** specifies the context for which the availability is requested.
- ***attributes*** **PAM_T_String[]** specifies the attributes of interest. Can be empty array to indicate all attributes.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***availability*** **PAM_T_AvailabilityProfile[]** containing a list of attributes as available to the asker in the requested context. If no information is available to the asker an array containing zero elements is returned.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All contexts may optionally include an asker profile. Although PAM applications may decide what attributes to include in an asker profile, PAM implementations should not require such attributes to be present. The implementations should leave it to the availability computations to decide the availability based on the (partial) information provided.

It is also up to the availability computation to decide on the trustworthiness of the asker profile information based on the application, the credentials of the entity asking for availability and/or the credentials, if any, of the entity accessing the interface.

PAM_AV_getPreference

Get the availability preferences of an identity for the specified communication mode.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity of interest.
- ***context*** **PAM_T_Context** specifies the context for which the preferences are requested.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***preference*** **PAM_T_Preference** the preference for the named capability if previously specified for the identity. Is null if there are no preferences associated.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

This method should be used in conjunction with the PAM_AV_setPreference method. It is not intended for the caller to implement preference-handling capabilities outside of the server with this method. Rather, this method is provided to query an existing preference value, for use in updating the same preferences with the setPreference method.

PAM_AV_setPreference

Set the availability preferences for the specified identity for the specified capability.

Inputs:

- ***identity*** **PAM_T_FQName** specifies the identity with which the preference will be associated.
- ***context*** **PAM_T_Context** specifies the capability to which this preference applies.
- ***newPreference*** **PAM_T_Preference** specifies the availability preference to add.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_IDENTITY** indicates that specified identity does not exist.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

Any existing preference for the same capability for the identity will be overwritten. No indication about the earlier existence of a preference is provided. To determine existing preferences when setting new preferences, use the PAM_AV_getPreference method.

If the new preference is specified as Null, any existing preferences will be removed.

3.7 Events

This Section describes the programmatic interface to Event notification service. The purpose of this interface is to manage the registrations of interest in events and the registration of client interfaces for subsequent notification.

All notifications in this specification are to be sent *after* the corresponding event has occurred and are asynchronous.

An application must first register a notification interface with the service. It can then register interest in one or more events for this interface.

A failure or a reset of a PAM implementation may result in a loss of all prior event and interface registrations. It is the responsibility of the client application to confirm the continued registration of the notification interface at regular intervals and re-register if necessary.

For security and privacy purposes, a registration for an event is allowed if and only if the supplied credentials during registration is sufficient to have allowed access to the information related to the event through one or more of the PAM interface methods.

Some implementations may choose not to implement the Event notification service.

3.7.1 Summary

Event registration methods:

PAM_EV_registerForEvent	register interest in an event
PAM_EV_deregisterFromEvent	deregister interest from an event

Notification interface registration methods:

PAM_EV_registerAppInterface	register the notification interface for an application
PAM_EV_deregisterAppInterface	deregister the notification interface of an application
PAM_EV_isRegistered	check if an interface is registered

PAM_EV_deregisterAppInterface

Deregister a client application’s notification interface.

Inputs:

- ***clientID*** **PAM_T_LongInteger** specifies the registration ID provided at registration.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_NOT_REGISTERED** a warning indicating that the interface was not previously registered.
- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All registrations for events for this client registration are also removed.

PAM_EV_deregisterFromEvent

Deregister a client application's interest in an event.

Inputs:

- ***eventID*** **PAM_T_LongInteger** specifies a prior event registration ID.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_NOT_REGISTERED** indicates that there was no registration corresponding to the supplied ID

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_EV_isRegistered

Check if a client application interface is registered.

Inputs:

- ***clientID*** **PAM_T_LongInteger** specifies the registration ID provided at registration.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***isRegistered*** **PAM_T_Boolean** is True if the registration ID is still valid, False otherwise.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_EV_registerAppInterface

Register a client application's notification interface.

Inputs:

- ***appInterface*** **PAM_T_Notification** specifies an implementation dependent handle to a client notification interface.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***clientID*** **PAM_T_LongInteger** is an ID returned by the service that uniquely identifies this registration.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

See Section 3.7.2 for the notification interface that the client application needs to provide.

PAM_EV_registerForEvent

Register a client application's interest in an event.

Inputs:

- ***clientID*** **PAM_T_LongInteger** specifies the notification interface registration.
- ***event*** **PAM_T_Event** specifies the event of interest.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***eventID*** **PAM_T_LongInteger** is an ID returned by the service that uniquely identifies this registration for the event.

Return Status:

- **PAM_NOT_REGISTERED** a warning indicating that a notification interface was not previously registered.
- **PAM_INVALID_EVENT** indicates that either the event is not defined or an attribute for the event is specified incorrectly.
- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

3.7.2 Application Notification Interface

This Section describes the interface that a client application must implement and register with the Event Service in order to be notified of events.

PAM_EA_eventNotify

Notify the occurrence of an event.

Inputs:

- ***eventID*** **PAM_T_LongInteger** specifies the event registration ID.
- ***eventInfo*** **PAM_T_EventInfo** contains the data about the event that occurred.
- ***authToken*** **PAM_T_Credential** of the PAM implementation.

Output:

- None.

Return Status:

- **PAM_SUCCESS** indicates that the notification was successful.
- **PAM_FAILURE** indicates that the notification failed for unspecified reasons.

Remarks:

The PAM implementations will not attempt to re-notify on failure.

3.8 Framework

The programmatic interface to framework services. The purpose of this interface is to provide the methods for initial contact and authentication of the client application prior to obtaining the PAM interfaces.

The method by which a client application gets access to the framework is implementation dependent. It could be through an URL, application brokers, etc.

Once the initial contact is possible, an application calls the PAM_FM_initAuthentication method to swap each other's authentication interfaces. The authentication interface is used to authenticate each other using a mutually agreeable authentication method.

Once authenticated, the application can call the PAM_FM_getAccess to get the handles to all of the PAM interface implementations.

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 PAM_FM_getAuthToken is used to obtain an implementation dependent token. An application that has authenticated itself with the 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. Implementations may provide expiration times for the authentication tokens.

3.8.1 Summary

Authentication and access methods:

PAM_FM_initAuthentication	Swap authentication interfaces
PAM_FM_getAccess	Get PAM interface handles after authentication
PAM_FM_getAuthToken	Get the credentials for a given askerData

Discovery methods:

PAM_FM_listServices	List available services

Identity namespace methods:

PAM_FM_getDefaultIdentityNamespace	return the namespace used for unique naming of identities in this PAM domain
PAM_FM_getDefaultAgentNamespace	return the namespace used

PAM_FM_initiateAuthentication

Swap authentication interfaces between the client and PAM implementation.

Inputs:

- ***clientID*** **PAM_T_String** specifies the identifier of the client. The framework may use this identifier to retrieve the public key for the client.
- ***clientAuthenticate*** **PAM_T_AuthenticationHandle** is an implementation-dependent handle to the client implementation of Authenticate interface.

Output:

- ***pamAuthenticate*** **PAM_T_AuthenticationHandle** is an implementation-dependent handle to the server implementation of Authenticate interface.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_UNAVAILABLE** indicates that the PAM implementation is unavailable at this time.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_FM_getAccess

Get access to the PAM implementation interfaces.

Inputs:

- ***serviceName*** **PAM_T_String** specifies the name of the service for which the access is requested.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***pamInterface*** **PAM_T_InterfaceHandle** is a implementation-dependent PAM service interface handle that all services extend.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_FM_getAuthToken

Get an authentication token for access to the interfaces.

Inputs:

- ***askerData*** **PAM_T_Data[]** specifies information about the asker. Can be an empty array.

Output:

- ***authToken*** **PAM_T_Credential** is an implementation-dependent authentication credential that can be verified.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the askerData information is not specified, the token corresponds to the entity that has authenticated to the framework is returned.

This method can be called any number of times for different asker profiles.

PAM_FM_getDefaultAgentNamespace

Get the namespace used for unique naming of agents in this PAM domain.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***defaultNamespace*** **PAM_T_String** contains name of the default namespace for agents in this PAM domain.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the namespace is not specified in any of the fully qualified names passed as parameters to the methods in this interface, the implementation assumes the default namespace.

PAM_FM_getDefaultIdentityNamespace

Get the namespace used for unique naming of identities in this PAM domain.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***defaultNamespace*** **PAM_T_String** contains name of the default namespace for identities in this PAM domain.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully..
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

If the namespace is not specified in any of the fully qualified names passed as parameters to the methods in this interface, the implementation assumes this default namespace.

PAM_FM_listServices

Get the list of service names supported by the server.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***services*** **PAM_T_String[]** contains the list of services supported by the server

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully..
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The strings returned correspond to the two letter codes used in the interface naming scheme (IM, AM, AA, AP, etc.)

3.8.2 Authenticate Interface

This Section describes the programmatic interface to the Authentication interface that PAM implementations and client applications implement to mutually authenticate each other.

Summary

PAM_FM_selectAuthMethod	select an authentication method from the list offered
PAM_FM_authenticate	authenticate using the selected method
PAM_FM_abortAuthentication	terminate authentication in progress

PAM_FM_abortAuthentication

Abort any authentication in progress.

Inputs:

- None.

Output:

- None.

Return Status:

- **PAM_SUCCESS** indicates that the authentication was aborted.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_FM_authenticate

Authenticates the server to the client.

Inputs:

- ***prescribedMethod*** **PAM_T_String** specifying the selected method of authentication.
- ***challenge*** **PAM_T_String** containing the challenge offered by the caller.

Output:

- ***response*** **PAM_T_String** containing a response to the challenge. May be encrypted by the mechanism used by the authentication method.

Return Status:

- **PAM_SUCCESS** indicates that the operation was successful.
- **PAM_UNABLE** indicates that the callee is unable to send a response to the challenge in the selected method.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_FM_selectAuthMethod

Select authentication method from the list of offered authentication capabilities.

Inputs:

- ***authCapabilities*** **PAM_T_String[]** specifies the list of authentication methods, the caller is capable of.

Output:

- ***prescribedMethod*** **PAM_T_String** containing the method selected by the callee from the list above.

Return Status:

- **PAM_SUCCESS** indicates that the selection was successful.
- **PAM_INCOMPATIBLE** indicates that the offered list of authentication method was not acceptable to the callee.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

4 Auxiliary Interfaces

4.1 Identity and Agent Types

This Section describes the programmatic interface to define the type schema for identities and agents specifying the attributes associated with the type. These types can then be assigned to agents and 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.

4.1.1 Summary

Identity Type methods:

PAM_TX_createIdentityAttribute	define an identity attribute
PAM_TX_deleteIdentityAttribute	remove an identity attribute definition
PAM_TX_getIdentityAttributeDefinition	get the identity attribute definition
PAM_TX_listAllIdentityAttributes	list all known identity attributes
PAM_TX_createIdentityType	create an identity type label
PAM_TX_deleteIdentityType	delete an identity type label
PAM_TX_listIdentityTypes	list all known identity types
PAM_TX_addIdentityTypeAttributes	add attributes to the identity type schema
PAM_TX_removeIdentityTypeAttributes	delete attributes from the identity type schema
PAM_TX_listIdentityTypeAttributes	list attributes in the identity type schema

Agent Type methods:

PAM_TX_createAgentAttribute	define an agent attribute
PAM_TX_deleteAgentAttribute	remove an agent attribute definition
PAM_TX_getAgentAttributeDefinition	get the agent attribute definition
PAM_TX_listAllAgentAttributes	list all known agent attributes
PAM_TX_createAgentType	create an agent type label
PAM_TX_deleteAgentType	delete an agent type label
PAM_TX_listAgentTypes	list all known agent types
PAM_TX_addAgentTypeAttributes	add attributes to the agent type schema
PAM_TX_removeAgentTypeAttributes	delete attributes from the agent type schema
PAM_TX_listAgentTypeAttributes	list attributes in the agent type schema

PAM_TX_addAgentTypeAttributes

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

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***attributeNames*** **PAM_T_String[]** list of attributes to be added to this type
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type does not exist.
- **PAM_ATTRIBUTE_EXISTS** indicates that at least one of the named attributes already exists
- **PAM_UNKNOWN_ATTRIBUTE** indicates that the at least one of the attributes specified has not been defined before

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_addIdentityTypeAttributes

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

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***attributeNames*** **PAM_T_String[]** list of attributes to be added to this type
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type does not exist.
- **PAM_ATTRIBUTE_EXISTS** indicates that at least one of the named attributes already exists
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the specified attributes has not been defined

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_createAgentAttribute

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

Inputs:

- ***attribute*** **PAM_T_AttributeDef** specifies the attribute to be created.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_ATTRIBUTE_EXISTS** indicates that the named attribute already exists.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_createAgentType

Specify a label as the name of an agent type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***attributeNames*** **PAM_T_String[]** list of attributes associated with this type. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_TYPE_EXISTS** indicates that the named type already exists.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes has not been defined

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_createIdentityAttribute

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

Inputs:

- ***attribute*** **PAM_T_AttributeDef** specifies the attribute to be created.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_ATTRIBUTE_EXISTS** indicates that the named attribute already exists.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_createIdentityType

Specify a label as the name of an identity type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type
- ***attributeNames*** **PAM_T_String[]** list of attributes associated with this type. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_TYPE_EXISTS** indicates that the named type already exists.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the specified attributes has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_deleteAgentAttribute

Delete the definition of an agent attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that the named attribute does not exist.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The removed attribute is no longer available as part of any agent type.

PAM_TX_deleteAgentType

Delete a label as the name of an agent type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_TX_deleteIdentityAttribute

Delete the definition of an identity attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that the named attribute does not exist.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The removed attribute is no longer available as part of any identity type.

PAM_TX_deleteIdentityType

Delete a label as the name of an identity type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

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.

PAM_TX_getAgentAttributeDefinition

Get the definition for the specified agent attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributeDef*** **PAM_T_AttributeDef** containing the definition of the attribute.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that no definition exists for the named attribute.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_getIdentityAttributeDefinition

Get the definition for the specified identity attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributeDef*** **PAM_T_AttributeDef** containing the definition of the attribute.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that no definition exists for the named attribute.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listAgentTypeAttributes

List all attributes of an agent type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **attributes** **PAM_T_AttributeDef[]** containing the list of attribute definitions for the named type. An empty array if no attributes have been defined for this type.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type has not been defined.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listAgentTypes

List all known agent types.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **typeNames** **PAM_T_String[]** containing the list of known agent types. An empty array if no agent types have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listAllAgentAttributes

List all known agent attributes.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **attributeNames** **PAM_T_String[]** containing the list of attribute names defined so far. An empty array if no attributes have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listAllIdentityAttributes

List all known identity attributes.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **attributeNames** **PAM_T_String[]** containing the list of attribute names defined so far. An empty array if no attributes have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listIdentityTypeAttributes

List all attributes of an identity type.

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **attributes** **PAM_T_AttributeDef[]** containing the list of attribute definitions for the named type. An empty array if no attributes have been defined for this type.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type has not been defined.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_listIdentityTypes

List all known identity types.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- **typeNames** **PAM_T_String[]** containing the list of known identity types. An empty array if no identity types have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_TX_removeAgentTypeAttributes

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

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***attributeNames*** **PAM_T_String[]** list of attribute names to be deleted from this type
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type does not exist.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes is not currently associated with the type.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All agent instances of this type are no longer associated with attributes that have been deleted.

PAM_TX_removeIdentityTypeAttributes

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

Inputs:

- ***typeName*** **PAM_T_String** specifies the name of the type.
- ***attributeNames*** **PAM_T_String[]** list of attribute names to be deleted from this type
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the named type does not exist.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes is not currently associated with the type.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All identity instances of this type are no longer associated with attributes that have been deleted.

4.2 Capabilities

This Section 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.

4.2.1 Summary

Capability methods:

PAM_CX_createCapabilityAttribute	define a capability attribute
PAM_CX_deleteCapabilityAttribute	remove a capability attribute definition
PAM_CX_getCapabilityAttributeDefinition	get the capability attribute definition
PAM_CX_listAllCapabilityAttributes	list all known capability attributes
PAM_CX_createCapability	create a capability label
PAM_CX_deleteCapability	delete a capability label
PAM_CX_listCapabilities	list all known capabilities
PAM_CX_addCapabilityAttributes	add attributes to the capability schema
PAM_CX_removeCapabilityAttributes	remove attributes from the capability schema
PAM_CX_listCapabilityAttributes	list attributes in the capability schema
PAM_CX_assignCapabilitiesToType	assign a list of capabilities to an agent type
PAM_CX_unassignCapabilitiesFromType	unassign capabilities from an agent type
PAM_CX_listCapabilitiesOfType	list capabilities assigned to an agent type

PAM_CX_addCapabilityAttributes

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

Inputs:

- ***capability*** **PAM_T_Capability** specifies the name of the capability.
- ***attributeNames*** **PAM_T_String[]** list of attributes to be added to this capability
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_CAPABILITY** indicates that the named capability does not exist.
- **PAM_ATTRIBUTE_EXISTS** indicates that at least one of the named attributes already exists
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the specified attributes has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_assignCapabilityToType

Assign capabilities to agent type.

Inputs:

- ***agentType*** **PAM_T_String** name of an agent type
- ***capability*** **PAM_T_Capability[]** specifies the list of capabilities to be associated.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the specified type does not exist.
- **PAM_UNKNOWN_CAPABILITY** indicates that at least one of the specified capabilities has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_createCapability

Specify a label as the name of a capability.

Inputs:

- ***capability*** **PAM_T_Capability** specifies the name of the capability.
- ***attributeNames*** **PAM_T_String[]** list of attributes associated with this capability. Can be an empty array.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_CAPABILITY_EXISTS** indicates that the named capability already exists.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the specified attributes has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_createCapabilityAttribute

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

Inputs:

- ***attribute*** **PAM_T_AttributeDef** specifies the attribute to be created.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_ATTRIBUTE_EXISTS** indicates that the named attribute already exists.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_deleteCapability

Delete a label as the name of a capability.

Inputs:

- ***capability*** **PAM_T_Capability** specifies the name of the capability to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_CAPABILITY** indicates that the named capability has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All agents that have this capability are no longer associated with this capability and consequently will no longer will have any attributes associated with this capability.

PAM_CX_deleteCapabilityAttribute

Delete the definition of a capability attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute to be deleted.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that the named attribute does not exist.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

The removed attribute is no longer available as part of any capability type.

PAM_CX_getCapabilityAttributeDefinition

Get the definition for the specified capability attribute.

Inputs:

- ***attributeName*** **PAM_T_String** specifies the attribute
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributeDef*** **PAM_T_AttributeDef** containing the definition of the attribute.

Return Status:

- **PAM_UNKNOWN_ATTRIBUTE** indicates that no definition exists for the named attribute.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_listCapabilities

List all known capabilities.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***capabilities*** **PAM_T_Capability[]** containing the list of known capability. An empty array if no capabilities have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_listCapabilitiesOfType

List capabilities assigned to an agent type.

Inputs:

- ***agentType*** **PAM_T_String** name of an agent type
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***capabilities*** **PAM_T_Capability[]** containing the list of capabilities assigned to the agent type

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the specified type does not exist.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_listAllCapabilityAttributes

List all known capability attributes.

Inputs:

- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributeNames*** **PAM_T_String[]** containing the list of attribute names defined so far. An empty array if no attributes have been defined.

Return Status:

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_listCapabilityAttributes

List all attributes of a capability.

Inputs:

- ***capability*** **PAM_T_Capability** specifies the name of the capability.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- ***attributes*** **PAM_T_AttributeDef[]** containing the list of attribute definitions for the named capability. An empty array if no attributes have been defined for this capability.

Return Status:

- **PAM_UNKNOWN_CAPABILITY** indicates that the named capability has not been defined.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

PAM_CX_removeCapabilityAttributes

Remove attribute definitions from the schema of capability that has already been defined.

Inputs:

- ***capability*** **PAM_T_Capability** specifies the name of the capability.
- ***attributeNames*** **PAM_T_String[]** list of attribute names to be deleted from this capability
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_CAPABILITY** indicates that the named capability does not exist.
- **PAM_UNKNOWN_ATTRIBUTE** indicates that at least one of the named attributes is not currently associated with the capability.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

All agent instances with this capability are no longer associated with attributes that have been removed.

PAM_CX_unassignCapabilityFromType

Unassign capabilities from an agent type.

Inputs:

- ***agentType*** **PAM_T_String** name of an agent type
- ***capability*** **PAM_T_Capability[]** specifies the list of capabilities to be disassociated.
- ***authToken*** **PAM_T_Credential** of the entity who wishes to do this operation.

Output:

- None.

Return Status:

- **PAM_UNKNOWN_TYPE** indicates that the specified type does not exist.
- **PAM_UNKNOWN_CAPABILITY** indicates that at least one of the specified capabilities has not been defined.

.

- **PAM_SUCCESS** indicates that the operation completed successfully.
- **PAM_FAILURE** indicates that the operation failed for unspecified reasons.
- **PAM_INVALID_CREDENTIAL** indicates that the credential presented is not recognized or insufficient for the operation.
- **PAM_NOT_SUPPORTED** implementation dependent status that indicates that this method is not supported by the implementation.

Remarks:

None.

5 PAM pre-defined objects

5.1 Pre-defined Constants

5.1.1 Communication modes

These are the constants to be used as values of the communication mode attributes. Currently, only the following are pre-defined as part of the PAM spec.

PAM_CM_IM	Instant Messaging
PAM_CM_VOICE	Voice calls
PAM_CM_SMS	SMS messaging

5.1.2 PAM interface codes

These are the constants to be used for getting access to PAM defined interfaces via the framework.

PAM_CI_IM	Identity Management
PAM_CI_AM	Agent Management
PAM_CI_AA	Agent Association
PAM_CI_AP	Agent Presence
PAM_CI_IP	Identity Presence
PAM_CI_AV	Availability Management
PAM_CI_EV	Event Management
PAM_CI_AX	Attribute Management
PAM_CI_CX	Capability Management
PAM_CI_TX	Type Management

5.1.3 Context names

These are constants used as names of contexts in availability queries.

PAM_CC_Communication	Communication context
PAM_CC_Location	Location context

5.1.4 Information privacy codes

These are the constants used as values of the privacy code attributes provided with the availability information.

PAM_CP_ASKER_ONLY	The information is for use by asker only.
PAM_CP_AUTHORIZED	The information can be propagated to authorized users only
PAM_CP_UNLIMITED	The information can be propagated to anyone.

5.2 Pre-defined Contexts

The contexts below are pre-defined for use with availability queries.

A context of type **PAM_T_Context**, in general, is a structure with the following attributes:

contextName PAM_T_String contains the name of the context

contextData PAM_T_Data[] contains a set of attributes based on the context

askerData PAM_T_Data[] contains a set of attributes describing the asker

An availability computation may use some or all of the data provided in the context to determine the availability information to be provided in response to a query. Except as may be specified in the context definitions below, all data provided in the context is optional.

5.2.1 Communication

This is a context for querying availability for communication.

contextName = PAM_T_Communication

contextData

communicationMode: PAM_T_String (see 5.1.1)

optional data…

askerData

askerIdentity; PAM_T_FQName Identity of the asker, if known in some namespace

name: PAM_T_String Any other identifying name of asker if identity is not known

optional asker data…

5.2.2 Location

This is a context for querying location availability.

contextName = PAM_T_Location

contextData

optional context data…

askerData

askerIdentity; PAM_T_FQName Identity of the asker, if known in some namespace

name: PAM_T_String Any other identifying name of asker if identity is not known

optional asker data…

5.3 Pre-defined Events

This section describes the events that are pre-defined by the PAM specifications as necessary to be supported by implementations.

5.3.1 Summary:

PAM_CE_IDENTITY_CREATED	Notify if an identity is deleted
PAM_CE_IDENTITY_DELETED	Notify if an identity is deleted
PAM_CE_ GROUP_MEMBERSHIP_CHANGED	Notify if the member list of a group changes
PAM_CE_AGENT_CREATED	Notify if a new agent has been created
PAM_CE_AGENT_DELETED	Notify if an agent is deleted
PAM_CE_AGENT_ASSIGNED	Notify if an agent is assigned to an identity
PAM_CE_AGENT_UNASSIGNED	Notify if an agent is unassigned from an identity
PAM_CE_CAPABILITY_CHANGED	Notify if an identity's capability changes
PAM_CE_IDENTITY_PRESENCE_CHANGED	Notify if an identity's presence data changes
PAM_CE_AGENT_PRESENCE_CHANGED	Notify if an agent's presence data changes
PAM_CE_AVAILABILITY_CHANGED	Notify if an identity's availability data changed

PAM_CE_IDENTITY_CREATED

Notify if a new identity has been created.

Required input attributes in the **PAM_T_Event** structure:

- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested. Can be an empty array if notification required for identities of all types.

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName[]** contains the names of the identities that have been created.

Remarks:

Notifications for creation of multiple identities are bunched into a single notification whenever possible.

A notification of this event is NOT sent for new association of types with an existing identity.

PAM_CE_IDENTITY_DELETED

Notify if an identity has been deleted.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the deletion is to be notified. Can be an empty array.
- ***identityType*** **PAM_T_String[]** specifies the types of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName[]** contains the names of the identities that have been deleted.

Remarks:

Notifications for deletion of multiple identities are bunched into a single notification whenever possible.

A notification of this event is NOT sent for removing association of types with an existing identity.

PAM_CE_GROUP_MEMBERSHIP_CHANGED

Notify if the membership of a group changes.

Required input attributes in the **PAM_T_Event** structure:

- ***groupName*** **PAM_T_FQName[]** specifies the name of the group for which the change is to be notified. Can be an empty array if notifications are required for any group.
- ***groupType*** **PAM_T_String[]** specifies the type of the group for which this notification is requested if the groupName is specified as an empty array. Can be an empty array if notification required for groups of any type.

Output attributes in the **PAM_T_EventInfo** structure:

- ***group*** **PAM_T_FQName[]** contains the names of the groups that have been changed.

Remarks:

Notifications for changes to multiple groups are bunched into a single notification whenever possible.

PAM_CE_AGENT_CREATED

Notify if a new agent has been created.

Required input attributes in the **PAM_T_Event** structure:

- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested. Can be an empty array if notification required for agents of any type.

Output attributes in the **PAM_T_EventInfo** structure:

- ***agents*** **PAM_T_FQName[]** contains the names of the agents that have been created.

Remarks:

Notifications for creation of multiple agents are bunched into a single notification whenever possible.

The notification for this event is NOT sent for new associations of types with agents.

PAM_CE_AGENT_DELETED

Notify if an agent has been deleted.

Required input attributes in the **PAM_T_Event** structure:

- ***agentName*** **PAM_T_FQName[]** specifies the name of the agent for which the deletion is to be notified. Can be an empty array.
- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested if the agentName is specified as an empty array. Can be an empty array if notification required for agents of any type.

Output attributes in the **PAM_T_EventInfo** structure:

- ***agents*** **PAM_T_FQName[]** contains the names of the agents that have been deleted.

Remarks:

Notifications for deletion of multiple agents are bunched into a single notification whenever possible.

This event notification is NOT sent for disassociating a type from an agent.

PAM_CE_AGENT_ASSIGNED

Notify if an agent is assigned to an identity.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the assignment is to be notified. Can be an empty array if notification is required for any identity instance.
- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.
- ***agentName*** **PAM_T_FQName[]** specifies the name of the agent for which the deletion is to be notified. Can be an empty array.
- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested if agentName is specified as an empty array. Can be an empty array if notification required for agents of any type

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName** contains the name of the identity to whom an agent has been assigned.
- ***agent*** **PAM_T_FQName** contains the name of the agent that has been assigned.

Remarks:

None.

PAM_CE_AGENT_UNASSIGNED

Notify if an agent has been unassigned from an identity.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the assignment is to be notified. Can be an empty array if notification is required for any identity instance.
- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.
- ***agentName*** **PAM_T_FQName[]** specifies the name of the agent for which the deletion is to be notified. Can be an empty array.
- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested if agentName is an empty array. Can be an empty array if notification required for agents of any type

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName** contains the name of the identity to whom an agent has been assigned.
- ***agent*** **PAM_T_FQName** contains the name of the agent that has been assigned.

Remarks:

None.

PAM_CE_CAPABILITY_CHANGED

Notify if the capability of an identity changes.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the assignment is to be notified. Can be an empty array if notification is required for any identity instance.
- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.
- ***capabilities*** **PAM_T_Capability[]** specifies capabilities of interest. Can be an empty array.

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName** contains the name of the identity whose capability has changed.
- ***capabilities*** **PAM_T_Capability[]** contains the capabilities that have changed (i.e., added or removed).

Remarks:

None.

PAM_CE_AGENT_CAPABILITY_PRESENCE_SET

Notify if the value of capability presence attributes of an agent is explicitly set.

Required input attributes in the **PAM_T_Event** structure:

- ***agentName*** **PAM_T_FQName[]** specifies the name of the agent for which the assignment is to be notified. Can be an empty array if notification is required for any agent instance.
- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested if the agentName is specified as an empty array. Can be an empty array if notification required for agents of any type.
- ***capabilities*** **PAM_T_Capability[]** specifies the capabilities of interest. Can be an empty array if notifications are required for any capability.
- ***attributeNames*** **PAM_T_String[]** specifies attributes of interest. Can be an empty array.

Output attributes in the **PAM_T_EventInfo** structure:

- ***agent*** **PAM_T_FQName** contains the name of the agent whose capability has changed.
- ***capability*** **PAM_T_Capability** contains the capability for which the attributes were set.
- ***attributeNames*** **PAM_T_String[]** contains the attribute names that have changed in value.

Remarks:

Expiration of the dynamic attributes does not trigger this notification.

PAM_CE_AGENT_PRESENCE_SET

Notify if the value of presence attributes of an agent is explicitly set.

Required input attributes in the **PAM_T_Event** structure:

- ***agentName*** **PAM_T_FQName[]** specifies the name of the agent for which the assignment is to be notified. Can be an empty array if notification is required for any agent instance.
- ***agentType*** **PAM_T_String[]** specifies the type of the agent for which this notification is requested if the agentName is specified as an empty array. Can be an empty array if notification required for agents of any type.
- ***attributeNames*** **PAM_T_String[]** specifies attributes of interest. Can be an empty array.

Output attributes in the **PAM_T_EventInfo** structure:

- ***agent*** **PAM_T_FQName** contains the name of the agent whose capability has changed.
- ***attributeNames*** **PAM_T_String[]** contains the attribute names that have changed in value.

Remarks:

Expiration of the dynamic attributes does not trigger this notification.

PAM_CE_IDENTITY_PRESENCE_SET

Notify if the value of presence attributes of an identity is explicitly set.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the assignment is to be notified. Can be an empty array if notification is required for any identity instance.
- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.
- ***attributeNames*** **PAM_T_String[]** specifies attributes of interest. Can be an empty array.

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName** contains the name of the identity whose capability has changed.
- ***attributeNames*** **PAM_T_String[]** contains the attribute names that have changed in value.

Remarks:

Expiration of the dynamic attributes do not trigger this notification.

PAM_CE_AVAILABILITY_CHANGED

Notify if the availability of an identity changes.

Required input attributes in the **PAM_T_Event** structure:

- ***identityName*** **PAM_T_FQName[]** specifies the name of the identity for which the assignment is to be notified. Can be an empty array if notification is required for any identity instance.
- ***identityType*** **PAM_T_String[]** specifies the type of the identity for which this notification is requested if the identityName is specified as an empty array. Can be an empty array if notification required for identities of any type.
- ***context*** **PAM_T_Context[]** specified the context in which the availability is to be monitored. Cannot be an empty array.
- ***attributeNames*** **PAM_T_String[]** specifies attributes of interest. Can be an empty array.

Output attributes in the **PAM_T_EventInfo** structure:

- ***identity*** **PAM_T_FQName** contains the name of the identity whose capability has changed.
- ***availability*** **PAM_T_AvailabilityInfo[]** contains the availability information that has changed.

Remarks:

None.

6 PAM datatypes

6.1 PAM_T_Attribute

A structure that represents an attribute.

Fields:

name **PAM_T_String** The Attribute’s name.

type **PAM_T_String** The Attribute’s type.

expiresIn **PAM_T_TimeInterval** The interval in milliseconds in which the attribute values are valid. A time interval of 0 indicates an expired attribute value. A time interval of PAM_MAX_LONGINT indicates static attribute values that never expire. Negative values indicate an expired value and the time for which it has expired.

values **PAM_T_Value[]** The values for the Attribute. This model allows multi-valued attributes. Cannot be an empty array.

6.2 PAM_T_AttributeDef

A structure to represent the definition of an attribute.

Fields:

name **PAM_T_String** The Attribute’s name

type **PAM_T_String** The Attribute’s type

isStatic **PAM_T_Boolean** True indicates that the attributes is always static and its values never expire. False indicates that the attribute can be dynamic and may contain values that expire.

isRevertOnExpiration **PAM_T_Boolean** True indicates that the attribute reverts to the default value on expiration. False indicates that the attribute will not revert to the default value.

defaultValues **PAM_T_Value[]** An attribute is always initialized with this value. If the *isRevertOnExpiration* attribute is set to true, a dynamic attribute that has expired while stored in a PAM implementation is reset to this value with the *expiresIn* interval set to PAM_MAX_LONGINT.

6.3 PAM_T_AuthenticationHandle

Implementation-dependent handle to an implementation of the Authenticate interface.

6.4 PAM_T_AvailabilityProfile

Contains a list of attribute values as determined by the definition of the context for which the availability is provided.

Fields:

privacyCode **PAM_T_String** contains the privacy code (See section 5.1.4)

availabilityData **PAM_T_Attribute**[] The attributes with availability information

6.5 PAM_T_Boolean

Synonymous with Boolean or equivalent data type in the implementation platform.

6.6 PAM_T_Byte

Mapped to an 8-bit data type in the implementation platform.

6.7 PAM_T_Capability

Synonymous with PAM_T_String.

6.8 PAM_T_Context

A structure to hold the data which defines the context in which an availability is queried. PAM pre-defines two contexts "*Communication*" and "*Location*".

Fields:

contextName **PAM_T_String**

Specifies the name of the context.

contextData **PAM_T_Data[]**

Contains the list of attributes that define the context. The attributes to be included for a given context are specified by the definition of the context.

askerData **PAM_T_Data[]**

Contains information about the asker of availability. The exact attributes in this profile are dependent on the application. PAM reserves the attribute "name" to contain the identity of the asker if known.

6.9 PAM_T_Credential

Represents credentials passed into the API methods.

Fields:

type **PAM_T_String**

Indicates what kind of authentication method this is.

credentialData **PAM_T_Byte[]**

An opaque octet array containing the credential.

6.10 PAM_T_Data

A structure that contains named data to be used as the value of various parameters in the interfaces.

Fields:

name	**PAM_T_String**	The name of the data.
type	**PAM_T_String**	The type of the data.
values	**PAM_T_Value**	The value for the data.

6.11 PAM_T_Event

Structure to define an event for notification registering.

Fields:

eventCode **: PAM_T_ShortInteger**

Identifies the event of interest. Whether certain minimal set of events should be pre-defined in the specification or it should be left entirely up to the implementation, is yet to be decided based on feedback to this draft.

eventData **: PAM_T_Data[]**

A list of attribute-value pairs where the attributes are determined by the definition of the event.

6.12 PAM_T_EventInfo

Structure that holds the data about an event that occurred to be sent in a notification.

Fields:

eventCode **: PAM_T_ShortInteger**

Identifies the event of interest. Whether certain minimal set of events should be pre-defined in the specification or it should be left entirely up to the implementation, is yet to be decided based on feedback to this draft.

eventInfo **: PAM_T_Data[]**

A list of attribute-value pairs where the attributes are determined by the definition of the event.

6.13 PAM_T_FQName

Structure that specifies a fully qualified name i.e., a name and a namespace.

Fields:

name	**: PAM_T_String**	The name.
namespace	**: PAM_T_String**	The namespace in which the name is unique.

6.14 PAM_T_Integer

Mapped to a 32-bit signed integer in the implementation platform.

6.15 PAM_T_InterfaceHandle

A platform-dependent abstract interface that all PAM services extend for their interfaces.

6.16 PAM_T_LongInteger

Mapped to a 64-bit signed integer in the implementation platform. The constant PAM_MAX_LONGINT stands for the largest positive value that is possible in the platform

6.17 PAM_T_Preference

Structure to specify preference data.

Fields:

format **: PAM_T_String**

The string specifies the format (or language) in which the preference in this structure is stored.

data **: PAM_T_Byte[]**

The data that represents the preference in the format specified.

6.18 PAM_T_ShortInteger

Mapped to a 16-bit signed integer in the implementation platform.

6.19 PAM_T_String

Synonymous with a **String** or equivalent data type in the implementation platform.

6.20 PAM_T_TimeInterval

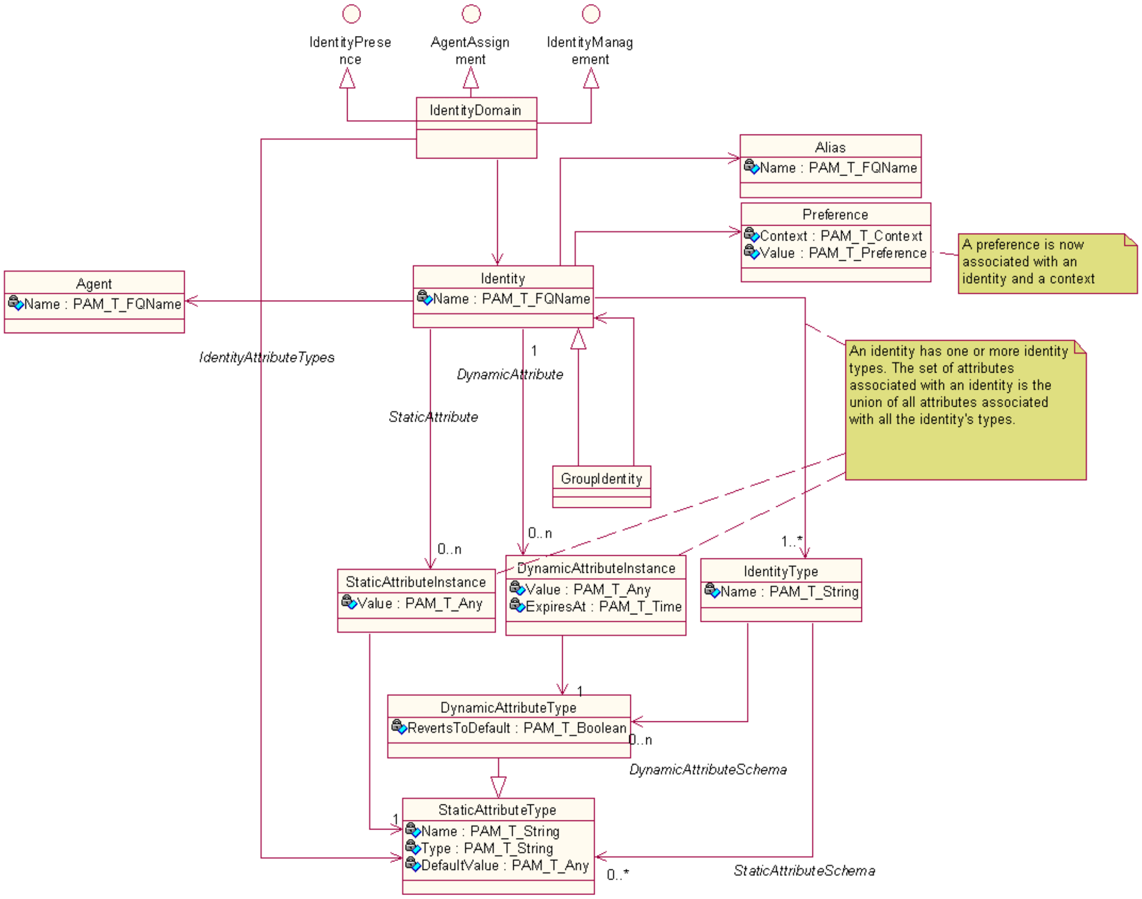
Synonymous with **PAM_T_LongInteger**.

6.21 PAM_T_Value

The data type of values is a discriminated union of the value types supported by the implementation.

7 Appendix A: UML models

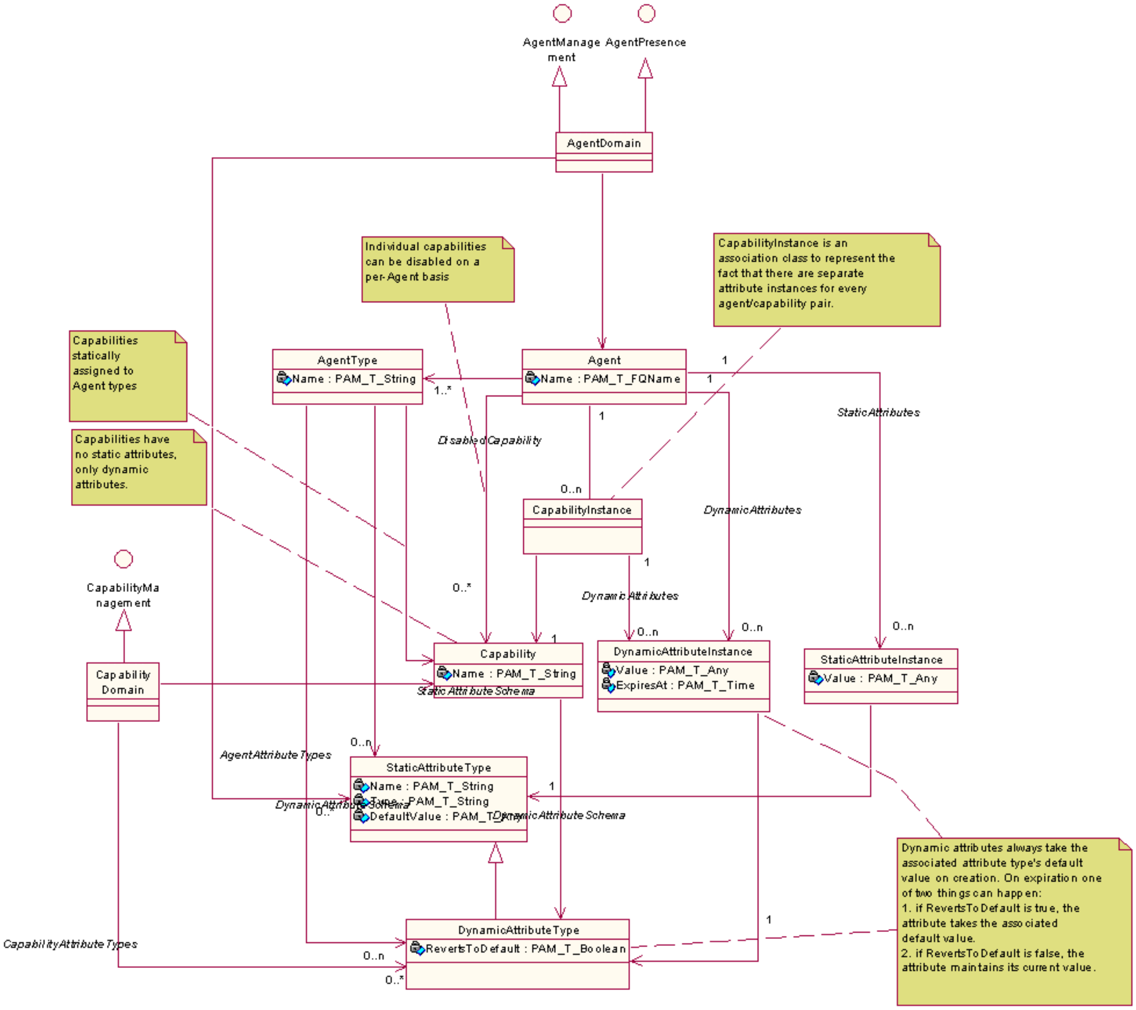
7.1 Identity



A preference is now associated with an identity and a context

An identity has one or more identity types. The set of attributes associated with an identity is the union of all attributes associated with all the identity's types.

7.2 Agent


AgentManagement
AgentPresence
AgentDomain
Individual capabilities can be disabled on a per-Agent basis
CapabilityInstance is an association class to represent the fact that there are separate attribute instances for every agent/capability pair.
Capabilities statically assigned to Agent types
Capabilities have no static attributes, only dynamic attributes.
AgentType
Name : PAM_T_String
Agent
Name : PAM_T_FQName
1..*
1
1
1
DisabledCapability
StaticAttributes
0..n
CapabilityInstance
DynamicAttributes
1
0..*
DynamicAttributes
0..n
0..n
0..n
1
CapabilityManagement
Capability
Name : PAM_T_String
Static Attribute Schema
DynamicAttributeInstance
Value : PAM_T_Any
ExpiresAt : PAM_T_Time
StaticAttributeInstance
Value : PAM_T_Any
Capability Domain
0..n
AgentAttributeTypes
StaticAttributeType
Name : PAM_T_String
Type : PAM_T_String
DefaultValue : PAM_T_Any
1
DynamicAttributeSchema
0..*
DynamicAttributeSchema
Dynamic attributes always take the associated attribute type's default value on creation. On expiration one of two things can happen:
1. if RevertsToDefault is true, the attribute takes the associated default value.
2. if RevertsToDefault is false, the attribute maintains its current value.
1
CapabilityAttributeTypes
DynamicAttributeType
RevertsToDefault : PAM_T_Boolean
0..n
0..*