AllJoyn™ Gateway Agent Framework Interface Definition

July 16, 2014

Contents

[1 Introduction 5](#_Toc396023828)

[1.1 Purpose 5](#_Toc396023829)

[1.2 Scope 5](#_Toc396023830)

[1.3 Release history 5](#_Toc396023831)

[1.4 References 5](#_Toc396023832)

[1.5 Acronyms and terms 5](#_Toc396023833)

[2 Specification 7](#_Toc396023834)

[2.1 Overview 7](#_Toc396023835)

[2.2 Discovery 8](#_Toc396023836)

[2.3 Discovery call flows 9](#_Toc396023837)

[2.3.1 Typical discovery flow 9](#_Toc396023838)

[3 Controller interfaces 10](#_Toc396023839)

[3.1 App Management interface 10](#_Toc396023840)

[3.1.1 Interface name 10](#_Toc396023841)

[3.1.2 Properties 10](#_Toc396023842)

[3.1.3 Methods 10](#_Toc396023843)

[3.1.4 AllJoyn Introspection XML 11](#_Toc396023844)

[3.2 App interface 11](#_Toc396023845)

[3.2.1 Interface name 11](#_Toc396023846)

[3.2.2 Properties 11](#_Toc396023847)

[3.2.3 Methods 11](#_Toc396023848)

[3.2.4 Signals 13](#_Toc396023849)

[3.2.5 AllJoyn Introspection XML 14](#_Toc396023850)

[3.3 Access control list management interface 15](#_Toc396023851)

[3.3.1 Interface name 15](#_Toc396023852)

[3.3.2 Properties 15](#_Toc396023853)

[3.3.3 Methods 15](#_Toc396023854)

[3.3.4 AllJoyn Introspection XML 17](#_Toc396023855)

[3.4 ACL interface 18](#_Toc396023856)

[3.4.1 Interface name 18](#_Toc396023857)

[3.4.2 Properties 18](#_Toc396023858)

[3.4.3 Methods 18](#_Toc396023859)

[3.4.4 AllJoyn Introspection XML 22](#_Toc396023860)

[4 Connector interfaces 24](#_Toc396023861)

[4.1 App interface 24](#_Toc396023862)

[4.1.1 Interface name 24](#_Toc396023863)

[4.1.2 Properties 24](#_Toc396023864)

[4.1.3 Methods 24](#_Toc396023865)

[4.1.4 Signals 25](#_Toc396023866)

[4.1.5 AllJoyn Introspection XML 25](#_Toc396023867)

Figures

[Figure 1. Gateway Agent framework context architecture 7](#_Toc396023868)

[Figure 2. Gateway Agent framework discovery 9](#_Toc396023869)

# Introduction

## Purpose

This document describes the specification of the AllJoyn™ Gateway Agent framework interfaces. These interfaces are required by an application to create and manage Access Control Lists (ACLs) that allow Controller applications that are connected to the cloud to communicate with other AllJoyn applications on the network.

## Scope

This document is targeted to the developers for AllJoyn applications.

## Release history

|  |  |
| --- | --- |
| Release version | What changed |
| 14.10 | The Gateway Agent framework's collection of interfaces were added:   * org.alljoyn.gwagent.ctrl.AppMgmt interface version 1 * org.alljoyn.gwagent.ctrl.App interface version 1 * org.alljoyn.gwagent.ctrl.AclMgmt interface version 1 * org.alljoyn.gwagent.ctrl.Acl interface version 1 * org.alljoyn.gwagent.connector.App interface version 1 |

## References

The following reference documents are found on the AllSeen Alliance web site's Docs and Downloads section.

* Introduction to the AllJoyn™ Framework
* AllJoyn™ Data Type Signature

■ AllJoyn™ About Feature 1.0 Interface Specification

The following reference document is found on the AllSeen Alliance wiki:

* AllJoyn™ Gateway Agent High-Level Design

## Acronyms and terms

| Acronym/Term | Definition |
| --- | --- |
| ACL | Access Control List |
| Announcement | A sessionless signal that exists as part of the About interface. Currently, once an announcement is triggered, it is sent periodically at 40sec interval. |
| Connector application | An application that communicates with proximal devices via alljoyn and is connected to the cloud to allow communication with proximal devices from afar. |
| Controller application | An application that is responsible for communicating with the GW Management App in order to define the Acls that will affect the Connector applications. |
| Gateway Management App | An application that is responsible for Acl Management for Connector applications. Connector applications are by default blocked from communicating with proximal devices for security reasons. By defining Acl’s Connectors can be given permissions to send/receive methods and signals for specific interfaces and/or object paths. |
| IoE | Internet of Everything |
| Exposed services | List of AllJoyn services provided by the Connector app which are being exposed for access by devices on the proximal network. |
| Remoted services | List of AllJoyn services that the Connector app can access if provided by remoted applications on the proximal network. |
| Remoted applications | List of AllJoyn apps which are being remoted (allowed for remote access). |

# Specification

## Overview

In a proximal AllJoyn IoE network, producer devices expose functionality-like notifications and control to other consumer devices in this network. This enables consumer devices to receive notifications about events or state changes on producer devices and display them to the user. This also enables user-initiated or machine-initiated control actions to be performed on producer devices, e.g., turning on/off the device, updating device settings etc. It is desirable to have a mechanism to *remote* such device functionality so that users can have the seamless experience of receiving notifications and/or controlling devices while away from the proximal network. This is also desirable for home automation use cases. The Gateway Agent framework is designed to provide such a mechanism.

Figure 1 shows context architecture for enabling remote access to services provided by devices in the AllJoyn network.



Figure . Gateway Agent framework context architecture

User sign up with a service provider to access AllJoyn device services remotely. The service provider enables remote services access via a Controller app that can be used to control devices or send/receive notifications. When in the proximal network, the Control app interacts with devices over the AllJoyn network. When outside the proximal network, the Controller app interacts with a cloud service hosted by the service provider.

A new Gateway Agent framework component is added in the AllJoyn network to enable remote access to AllJoyn applications outside the network. The Gateway Agent framework has a single Gateway Management app and one or more Connector apps. These AllJoyn-enabled applications interact with a single preinstalled AllJoyn router on the Gateway Agent framework. The Gateway Management app enables user to define and manage remote profiles via the Controller app. A remote profile lists the set of devices/apps/interfaces for which the user can enable the remote access. The remote access to AllJoyn-enabled devices is controlled via the config file at the AllJoyn router. The Gateway Management app is responsible for updating the config file to only allow access to remoted AllJoyn-enabled devices. Remote profile information is exposed to Connector apps via an AllJoyn interface provided by the Gateway Management app.

The Connector app maintains a long-lived connection with the service provider's cloud service. This persistent connection is used to receive control actions initiated by the user remotely on the Control app or by the cloud service itself (e.g., in case of home automation). The communication between the Connector app and the cloud service is through the service provider's defined web protocol. This is outside the scope of this document.

The Connector app also acts as a proxy for propagating notifications from the proximal network to the cloud service and vice versa.

When outside the proximal network, the Controller app interacts with a cloud service hosted by the service provider, which interacts with the Connector app on the Gateway Agent framework. The Connector app provide protocol conversion from service provider’s protocol to the AllJoyn-based protocol, and invokes interaction with remoted interfaces on the devices as described above.

## Discovery

The Controller app looks for a Gateway Agent framework that has a Gateway Management app supporting the Gateway app management interface. Figure 2 illustrates the following use cases:

* The Controller app is looking for a specific Gateway Agent framework identified by the AppName in the announcement signal from the Gateway Management app, for example, Service Provider 1 Gateway Agent.
* The Controller app wants to discover any Gateway Agent framwork.

## Discovery call flows

### Typical discovery flow

Figure 2 illustrates a typical call flow for a client Controller App to discover a Gateway Management App. The client merely relies on the announcement to decide whether to connect to the service app to use its service offering.



Figure . Gateway Agent framework discovery

# Controller interfaces

This chapter defines the interfaces used by the Controller application.

## App Management interface

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Object path |
| org.alljoyn.gwagent.ctrl.AppMgmt | 1 | yes | /gw |

### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Signature | List of values | Writable | Description |
| Version | q | Positive integers | no | Interface version number |

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.gwagent.ctrl.AppMgmt interface.

#### GetInstalledApps

Inputs

None.

Output

| Return signature | Parameter name | Mandatory | Description |
| --- | --- | --- | --- |
| a(ssos) | ConnectorId | yes | An array of structs containing information for the Installed app.  ID of the installed Connector app. |
| AppFriendlyName | yes | Friendly name for the application. |
| objectPath | yes | Object path used for the application. |
| Version | yes | Version of the application. |

Description

Returns list of already installed Connector apps on the Gateway Agent framework.

### AllJoyn Introspection XML

The following XML defines the org.alljoyn.gwagent.ctrl.AppMgmt interface.

<node name="/gw"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.gwagent.ctrl.AppMgmt">

<method name="GetInstalledApps">

<arg name="installedAppsInfoArray" type="a(ssos)" direction="out"/>

</method>

<property name="Version" type="q" access="read"/>

<annotation name="org.alljoyn.Bus.Secure" value="true"/>

</interface>

</node>

## App interface

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Suggested object path |
| org.alljoyn.gwagent.ctrl.App | 1 | yes | /gw/ConnectorId |

### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Signature | List of values | Writable | Description |
| Version | q | Positive integers | no | Interface version number |

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn. gwagent.ctrl.App interface.

#### GetAppStatus

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| qsqq | installStatus | GW\_IS\_INSTALLED = 0  GW\_IS\_INSTALL\_IN\_PROGRESS = 1  GW\_IS\_UPGRADE\_IN\_PROGRESS = 2  GW\_IS\_UNINSTALL\_IN\_PROGRESS = 3  GW\_IS\_INSTALL\_FAILED = 4  GW\_IS\_MAX\_INSTALL\_STATUS | yes | Install status of the application. |
| installDescription | N/A | no | Description of the installation. |
| connectionStatus | GW\_CS\_NOT\_INITIALIZED = 0  GW\_CS\_IN\_PROGRESS = 1  GW\_CS\_CONNECTED = 2  GW\_CS\_NOT\_CONNECTED = 3  GW\_CS\_ERROR = 4  GW\_CS\_MAX\_CONNECTION\_STATUS | yes | Connection status of the application. |
| operationalStatus | GW\_OS\_RUNNING = 0  GW\_OS\_STOPPED = 1  GW\_OS\_MAX\_OPERATIONAL\_STATUS | yes | Operational status of the application. |

Description

Method to receive information about the state of the application.

#### RestartApp

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | restartResponseCode | GW\_RESTART\_APP\_RC\_SUCCESS = 0  GW\_RESTART\_APP\_RC\_INVALID = 1 | yes | Code that signals success or failure of restart. |

Description

A method used to restart the application.

#### GetManifestFile

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| s | manifestFile | Dependent on implementation | yes | Contents of the manifest file. |

Description

Return the contents of the manifest file for the application. Refer to the AllJoyn™ Gateway Agent High-Level Design document for more information.

#### GetManifestInterfaces

Inputs

None.

Output

| Return signature | Parameter name | Mandatory | Description |
| --- | --- | --- | --- |
| a((obs)a(ssb)) | exposedServices | no | Exposed services defined in the manifest file. |
| a((obs)a(ssb)) | remotedServices | no | Remoted services defined in the manifest file. |

Description

Get the list of exposed services and remoted services, provided by the application, found in the manifest file.

### Signals

| Signal name | Parameter name | | | Sessionless | Description |
| --- | --- | --- | --- | --- | --- |
| AppStatusChanged | Name | Signature | Mandatory | no | Tells Controller applications that one of the application’s status fields changed.  The following information is provided in the signal:   * installStatus—the installation status of the application * installDescription—The Description if installation failed * connectionStatus – The connection Status of the application towards the cloud * operationalStatus – Status reflecting whether the application is running or not |
| installStatus | q | yes |
| installDescription | s | no |
| connectionStatus | q | yes |
| operationalStatus | q | yes |

### AllJoyn Introspection XML

The following XML defines the org.alljoyn.gwagent.ctrl.App interface.

<node name="/gw/ConnectorId"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.gwagent.ctrl.App">

<signal name="AppStatusChanged">

<arg name="installStatus" type="q" direction="out"/>

<arg name="installDescription" type="s" direction="out"/>

<arg name="connectionStatus" type="q" direction="out"/>

<arg name="operationalStatus" type="q" direction="out"/>

</signal>

<method name="GetAppStatus">

<arg name="installStatus" type="q" direction="out"/>

<arg name="installDescription" type="s" direction="out"/>

<arg name="connectionStatus" type="q" direction="out"/>

<arg name="operationalStatus" type="q" direction="out"/>

</method>

<method name="GetManifestFile">

<arg name="manifestFile" type="s" direction="out"/>

</method>

<method name="GetManifestInterfaces">

<arg name="exposedServices" type="a((obs)a(ssb))" direction="out"/>

<arg name="remotedServices" type="a((obs)a(ssb))" direction="out"/>

</method>

<method name="RestartApp">

<arg name="restartResponseCode" type="q" direction="out"/>

</method>

<property name="Version" type="q" access="read"/>

<annotation name="org.alljoyn.Bus.Secure" value="true"/>

</interface></node>

## Access control list management interface

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Suggested object path |
| org.alljoyn.gwagent.ctrl.AclMgmt | 1 | yes | /gw/ConnectorId |

### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Signature | List of values | Writable | Description |
| Version | q | Positive integers | no | Interface version number |

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.gwagent.ctrl.AclMgmt interface.

#### CreateAcl

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| aclName | yes | S | N/A | Name of the ACL. |
| exposedServices | no | a(obas) | N/A | List of exposed services that will be permitted as part of this ACL definition. |
| remotedApps | no | a(saya(obas)) | N/A | List of remoted applications and their object paths/interfaces that the Connector app will be permitted to communicate with as part of this ACL definition. |
| metadata | no | a{ss} | N/A | Metadata pertaining to this ACL – will be sent and maintained by Controller Management App. |
| customMetadata | no | a{ss} | N/A | Additional metadata pertaining to this ACL – can be set and maintained by the Controller Application layer. |

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | no | Success/failure of the ACL creation. |
| S | aclId | N/A | no | ID of the created ACL. |
| o | objectPath | N/A | no | Object path of the newly created ACL. |

Description

Method used to create an ACL.

#### DeleteAcl

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| aclId | No | s | N/A | ID of the ACL to delete. |

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| Q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure of the ACL deletion. |

Description

Method used to delete an existing ACL.

#### ListAcls

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| a(ssqo) | AclsList | N/A | yes | Array of ACLs defined for this application.  For each ACL, the aclId, aclName, aclStatus, and objectPath are returned. |

Description

Method used to receive a list of all ACLs defined for the application.

### AllJoyn Introspection XML

The following XML defines the org.alljoyn.gwagent.ctrl.AclMgmt interface.

<node name="/gw/ConnectorId"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.gwagent.ctrl.AclMgmt">

<method name="CreateAcl">

<arg name="aclName" type="s" direction="in"/>

<arg name="exposedServices" type="a(obas)" direction="in"/>

<arg name="remotedApps" type="a(saya(obas))" direction="in"/>

<arg name="metadata" type="a{ss}" direction="in"/>

<arg name="customMetadata" type="a{ss}" direction="in"/>

<arg name="aclResponseCode" type="q" direction="out"/>

<arg name="aclId" type="s" direction="out"/>

<arg name="objectPath" type="o" direction="out"/> </method>

<method name="DeleteAcl">

<arg name="aclId" type="s" direction="in"/>

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<method name="ListAcls">

<arg name="aclsList" type="a(sso)" direction="out"/>

</method>

<property name="Version" type="q" access="read"/>

<annotation name="org.alljoyn.Bus.Secure" value="true"/>

</interface>

</node>

## ACL interface

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Suggested object path |
| org.alljoyn.gwagent.ctrl.Acl | 1 | yes | /gw/ConnectorId/AclId |

### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Signature | List of values | Writable | Description |
| Version | q | Positive integers | no | Interface version number |

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn.gwagent.ctrl.Acl interface.

#### ActivateAcl

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure of activating an ACL. |

Description

Method used to active an ACL.

#### DeactivateAcl

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure deactivating an ACL. |

Description

Method used to deactivate an ACL.

#### GetAcl

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| s | aclName | N/A | yes | ACL name. |
| a(obas) | exposedServices | N/A | no | Exposed services used with this ACL. |
| a(saya(obas)) | RemotedApps | N/A | no | Remoted applications used with this ACL. |
| a(ss) | Metadata | N/A | no | Metadata used with this ACL. |
| a(ss) | CustomMetadata | N/A | no | Custom metadata used with this ACL. |

Description

Method used to get the contents of an ACL.

#### GetAclStatus

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclStatus | GW\_AS\_INACTIVE = 0  GW\_AS\_ACTIVE = 1 | yes | The status of the ACL. |

Description

Method used to get the status of an ACL.

#### UpdateAcl

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| aclName | yes | s | N/A | Name of the ACL. |
| exposedServices | no | a(obas) | N/A | List of services provided by the Connector app that are being exposed for access by devices on the proximal network and is permitted as part of this ACL definition. |
| remotedApps | no | a(saya(obas)) | N/A | List of remoted applications and their object paths/interfaces that the Connector app is permitted to communicate with as part of this ACL definition. |
| metadata | no | a{ss} | N/A | Metadata pertaining to this ACL – will be sent and maintained by Controller Service layer. |
| customMetadata | no | a{ss} | N/A | Additional metadata pertaining to this ACL – can be set and maintained by the Controller Application layer. |

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure of an ACL update. |

Description

Method used to update an ACL.

#### UpdateAclMetadata

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| metadata | no | a{ss} | N/A | Metadata pertaining to this ACL – will be sent and maintained by Controller Service layer. |

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure of the metadata update. |

Description

Method used to update the metadata of an ACL.

#### UpdateAclCustomMetadata

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| customMetadata | no | a{ss} | N/A | Additional metadata pertaining to this ACL – can be set and maintained by the Controller Application layer. |

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| q | aclResponseCode | GW\_ACL\_RC\_SUCCESS = 0  GW\_ACL\_RC\_INVALID = 1  GW\_ACL\_RC\_REGISTER\_ERROR = 2  GW\_ACL\_RC\_ACL\_NOT\_FOUND = 3  GW\_ACL\_RC\_PERSISTENCE\_ERROR = 4  GW\_ACL\_RC\_POLICYMANAGER\_ERROR = 5  GW\_ACL\_RC\_METADATA\_ERROR = 6 | yes | Success/failure of the metadata update. |

Description

Method used to update custom metadata of an ACL.

### AllJoyn Introspection XML

The following XML defines the org.alljoyn.gwagent.ctrl.Acl interface.

<node name="/gw/ConnectorId/AclId"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.gwagent.ctrl.Acl">

<method name="ActivateAcl">

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<method name="DeactivateAcl">

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<method name="GetAcl">

<arg name="aclName" type="s" direction="out"/>

<arg name="exposedServices" type="a(obas)" direction="out"/>

<arg name="remotedApps" type="a(saya(obas))" direction="out"/>

<arg name="metadata" type="a{ss}" direction="out"/>

<arg name="customMetadata" type="a{ss}" direction="out"/>

</method>

<method name="GetAclStatus">

<arg name="aclStatus" type="q" direction="out"/>

</method>

<method name="UpdateAcl">

<arg name="aclName" type="s" direction="in"/>

<arg name="exposedServices" type="a(obas)" direction="in"/>

<arg name="remotedApps" type="a(saya(obas))" direction="in"/>

<arg name="metadata" type="a{ss}" direction="in"/>

<arg name="customMetadata" type="a{ss}" direction="in"/>

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<method name="UpdateAclMetadata">

<arg name="metadata" type="a{ss}" direction="in"/>

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<method name="UpdateAclCustomMetadata">

<arg name="metadata" type="a{ss}" direction="in"/>

<arg name="aclResponseCode" type="q" direction="out"/>

</method>

<property name="Version" type="q" access="read"/>

<annotation name="org.alljoyn.Bus.Secure" value="true"/>

</interface>

</node>

# Connector interfaces

This chapter defines the Connector interfaces used by the Connector application.

## App interface

### Interface name

|  |  |  |  |
| --- | --- | --- | --- |
| Interface name | Version | Secured | Suggested object path |
| org.alljoyn.gwagent.connector.App | 1 | yes | /gw/ConnectorId |

### Properties

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Property name | Signature | List of values | Writable | Description |
| Version | q | Positive integers | no | Interface version number |

### Methods

The following methods are exposed by a BusObject that implements the org.alljoyn. gwagent.connector.App interface.

#### GetMergedAcl

Inputs

None.

Output

| Return signature | Parameter name | List of values | Mandatory | Description |
| --- | --- | --- | --- | --- |
| a(obas) | exposedServices | N/A | no | Returns the exposedServices defined in all active ACLs for this application. |
| a(saya(obas)) | remotedApps | N/A | no | Returns the remotedApps defined in all active ACLs for this application. |

Description

Method to receive information about the active ACLs of the application.

#### UpdateConnectionStatus

Inputs

| Parameter name | Mandatory | Signature | List of values | Description |
| --- | --- | --- | --- | --- |
| ConnectionStatus | yes | q | GW\_CS\_NOT\_INITIALIZED = 0  GW\_CS\_IN\_PROGRESS = 1  GW\_CS\_CONNECTED = 2  GW\_CS\_NOT\_CONNECTED = 3  GW\_CS\_ERROR = 4  GW\_CS\_MAX\_CONNECTION\_STATUS | Current connectionStatus of the Connector application. |

Output

None.

Description

Method used to update GatewayManagementApp with the current connectionStatus of the Connector app.

### Signals

| Signal name | Mandatory | Parameter name | Sessionless | Description |
| --- | --- | --- | --- | --- |
| AclUpdated | no | aclId | no | Tells Connector applications that an ACL was created, deleted, updated, activated, or deactivated. |
| ShutdownApp | yes | None | no | Used by the GatewayManagementApp to tell the Connector application that it should shut down. |

### AllJoyn Introspection XML

The following XML defines the org.alljoyn.gwagent.connector.App interface.

<node name="/gw/ConnectorId"

[xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"](http://www.w3.org/2001/XMLSchema-instance)

[xsi:noNamespaceSchemaLocation="http://www.allseenalliance.org/schemas/introspec](http://www.alljoyn.org/schemas/introspec) t.xsd">

<interface name="org.alljoyn.gwagent.connector.App">

<method name="GetMergedAcl">

<arg name="exposedServices" type="a(obas)" direction="out"/>

<arg name="remotedApps" type="a(saya(obas))" direction="out"/>

</method>

<signal name="AclUpdated">

<arg name="aclId" type="s" direction="out"/>

</signal>

<signal name="ShutdownApp">

</signal>

<method name="UpdateConnectionStatus">

<arg name="connectionStatus" type="q" direction="in"/>

<annotation name="org.freedesktop.DBus.Method.NoReply" value="true"/>

</method>

<annotation name="org.alljoyn.Bus.Secure" value="true"/>

</interface>

</node>