

**Open Service Access (OSA);  
Parlay X Web Services;  
Part 13: Address List Management  
(Parlay X 2)**



---

Reference

RES/TISPAN-01056-13-OSA

---

Keywords

API, OSA, service

**ETSI**

650 Route des Lucioles  
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C  
Association à but non lucratif enregistrée à la  
Sous-Préfecture de Grasse (06) N° 7803/88

---

**Important notice**

Individual copies of the present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in more than one electronic version or in print. In any case of existing or perceived difference in contents between such versions, the reference version is the Portable Document Format (PDF). In case of dispute, the reference shall be the printing on ETSI printers of the PDF version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced except as authorized by written permission.  
The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2008.

© The Parlay Group 2008.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™**, **TIPHON™**, the TIPHON logo and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members.

**3GPP™** is a Trade Mark of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

# Contents

Intellectual Property Rights .....	5
Foreword.....	5
1 Scope .....	6
2 References .....	6
2.1 Normative references .....	6
3 Definitions and abbreviations.....	7
3.1 Definitions.....	7
3.2 Abbreviations .....	7
4 Detailed service description .....	7
4.1 Group URI format .....	7
4.2 Address list usage in services .....	8
5 Namespaces.....	8
6 Sequence diagrams .....	9
6.1 Manage groups (Create, delete, query, set access and query access) .....	9
6.2 Manage group members (AddMember, AddMembers, DeleteMember, DeleteMembers, QueryMembers) .....	10
7 XML Schema data type definition .....	10
7.1 AccessPermissions structure .....	10
7.2 AttributeStatus enumeration.....	10
7.3 SimpleAttribute structure .....	11
8 Web Service interface definition.....	11
8.1 Interface: GroupManagement.....	11
8.1.1 Operation: createGroup.....	11
8.1.1.1 Input message: createGroupRequest .....	11
8.1.1.2 Output message: createGroupResponse .....	12
8.1.1.3 Referenced faults.....	12
8.1.2 Operation: deleteGroup.....	12
8.1.2.1 Input message: deleteGroupRequest .....	12
8.1.2.2 Output message: deleteGroupResponse .....	12
8.1.2.3 Referenced faults.....	12
8.1.3 Operation: queryGroups .....	12
8.1.3.1 Input message: queryGroupsRequest .....	13
8.1.3.2 Output message: queryGroupsResponse .....	13
8.1.3.3 Referenced faults.....	13
8.1.4 Operation: setAccess.....	13
8.1.4.1 Input message: setAccessRequest .....	14
8.1.4.2 Output message: setAccessResponse .....	14
8.1.4.3 Referenced faults.....	14
8.1.5 Operation: queryAccess .....	14
8.1.5.1 Input message: queryAccessRequest.....	14
8.1.5.2 Output message: queryAccessResponse .....	14
8.1.5.3 Referenced faults.....	14
8.2 Interface: Group .....	15
8.2.1 Operation: addMember .....	15
8.2.1.1 Input message: addMemberRequest.....	15
8.2.1.2 Output message: addMemberResponse.....	15
8.2.1.3 Referenced faults.....	15
8.2.2 Operation: addMembers .....	15
8.2.2.1 Input message: addMembersRequest .....	15
8.2.2.2 Output message: addMembersResponse .....	15
8.2.2.3 Referenced faults.....	16
8.2.3 Operation: deleteMember .....	16
8.2.3.1 Input message: deleteMemberRequest .....	16

8.2.3.2	Output message: deleteMemberResponse .....	16
8.2.3.3	Referenced faults.....	16
8.2.4	Operation: deleteMembers.....	16
8.2.4.1	Input message: deleteMembersRequest .....	16
8.2.4.2	Output message: deleteMembersResponse .....	17
8.2.4.3	Referenced faults.....	17
8.2.5	Operation: queryMembers .....	17
8.2.5.1	Input message: queryMembersRequest.....	17
8.2.5.2	Output message: queryMembersResponse.....	17
8.2.5.3	Referenced faults.....	17
8.2.6	Operation: addGroupAttribute .....	18
8.2.6.1	Input message: addGroupAttributeRequest.....	18
8.2.6.2	Output message: addGroupAttributeResponse.....	18
8.2.6.3	Referenced faults.....	18
8.2.7	Operation: deleteGroupAttribute .....	18
8.2.7.1	Input message: deleteGroupAttributeRequest .....	18
8.2.7.2	Output message: deleteGroupAttributeResponse.....	18
8.2.7.3	Referenced faults.....	18
8.2.8	Operation: queryGroupAttributes .....	19
8.2.8.1	Input message: queryGroupAttributesRequest.....	19
8.2.8.2	Output message: queryGroupAttributesResponse.....	19
8.2.8.3	Referenced faults.....	19
8.2.9	Operation: addGroupMemberAttribute.....	19
8.2.9.1	Input message: addGroupMemberAttributeRequest .....	19
8.2.9.2	Output message: addGroupMemberAttributeResponse .....	19
8.2.9.3	Referenced faults.....	19
8.2.10	Operation: deleteGroupMemberAttribute .....	20
8.2.10.1	Input message: deleteGroupMemberAttributeRequest .....	20
8.2.10.2	Output message: deleteGroupMemberAttributeResponse .....	20
8.2.10.3	Referenced faults.....	20
8.2.11	Operation: queryGroupMemberAttributes.....	20
8.2.11.1	Input message: queryGroupMemberAttributesRequest .....	20
8.2.11.2	Output message: queryGroupMemberAttributesResponse .....	20
8.2.11.3	Referenced faults.....	20
8.3	Interface: Member .....	21
8.3.1	Operation: addMemberAttribute.....	21
8.3.1.1	Input message: addMemberAttributeRequest .....	21
8.3.1.2	Output message: addMemberAttributeResponse .....	21
8.3.1.3	Referenced faults.....	21
8.3.2	Operation: queryMemberAttributes .....	21
8.3.2.1	Input message: queryMemberAttributesRequest .....	21
8.3.2.2	Output message: queryMemberAttributesResponse .....	21
8.3.2.3	Referenced faults.....	22
8.3.3	Operation: deleteMemberAttribute .....	22
8.3.3.1	Input message: deleteMemberAttributeRequest.....	22
8.3.3.2	Output message: deleteMemberAttributeResponse .....	22
8.3.3.3	Referenced faults.....	22
9	Fault definitions.....	22
9.1	PolicyException .....	22
9.1.1	POL0210: Too many members in group.....	22
9.1.2	POL0211: Subgroups not supported.....	23
9.1.3	POL0212: Group name too long.....	23
9.1.4	POL0213: Group already exists.....	23
10	Service policies .....	23
<b>Annex A (normative):</b>	<b>WSDL for Address List Management.....</b>	<b>24</b>
<b>Annex B (informative):</b>	<b>Bibliography.....</b>	<b>25</b>
History .....		26

---

## Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (<http://webapp.etsi.org/IPR/home.asp>).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

---

## Foreword

This ETSI Standard (ES) has been produced by ETSI Technical Committee Telecommunications and Internet converged Services and Protocols for Advanced Networking (TISPAN).

The present document is part 13 of a multi-part deliverable covering Open Service Access (OSA); Parlay X Web Services, as identified below:

- Part 1: "Common";
- Part 2: "Third Party Call";
- Part 3: "Call Notification";
- Part 4: "Short Messaging";
- Part 5: "Multimedia Messaging";
- Part 6: "Payment";
- Part 7: "Account Management";
- Part 8: "Terminal Status";
- Part 9: "Terminal Location";
- Part 10: "Call Handling";
- Part 11: "Audio Call";
- Part 12: "Multimedia Conference";
- Part 13: "Address List Management";**
- Part 14: "Presence".

The present document has been defined jointly between ETSI, The Parlay Group (<http://www.parlay.org>) and the 3GPP.

**The present document forms part of the Parlay X 2.2 set of specifications.**

**The present document is equivalent to 3GPP TS 29.199-13 V6.5.0 (Release 6).**

---

# 1 Scope

The present document is part 13 of the Stage 3 Parlay X 2 Web Services specification for Open Service Access (OSA).

The OSA specifications define an architecture that enables application developers to make use of network functionality through an open standardized interface, i.e. the OSA APIs.

The present document specifies the Address List Management Web Service. The following are defined here:

- Name spaces.
- Sequence diagrams.
- Data definitions.
- Interface specification plus detailed method descriptions.
- Fault definitions.
- Service Policies.
- WSDL Description of the interfaces.

---

# 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific.

- For a specific reference, subsequent revisions do not apply.
- Non-specific reference may be made only to a complete document or a part thereof and only in the following cases:
  - if it is accepted that it will be possible to use all future changes of the referenced document for the purposes of the referring document;
  - for informative references.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

For online referenced documents, information sufficient to identify and locate the source shall be provided. Preferably, the primary source of the referenced document should be cited, in order to ensure traceability. Furthermore, the reference should, as far as possible, remain valid for the expected life of the document. The reference shall include the method of access to the referenced document and the full network address, with the same punctuation and use of upper case and lower case letters.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

## 2.1 Normative references

The following referenced documents are indispensable for the application of the present document. For dated references, only the edition cited applies. For non-specific references, the latest edition of the referenced document (including any amendments) applies.

[1] W3C Recommendation (2 May 2001): "XML Schema Part 2: Datatypes".

NOTE: Available at <http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/>.

- [2] ETSI ES 202 391-1: "Open Service Access (OSA); Parlay X Web Services; Part 1: Common (Parlay X 2)".
- [3] IETF RFC 2396: "Uniform Resource Identifiers (URI): Generic Syntax".

---

## 3 Definitions and abbreviations

### 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 202 391-1 [2] and the following apply:

**application managed group:** group created and managed outside of the network, requiring the group members to be passed into the network for processing

**group:** container for a set of addresses, it is not an address itself. When a group contain one or more groups, logically the group contains the set of addresses it holds, plus the set of addresses that any contained group holds (including any addresses contained in groups that a contained group holds)

**group resolution:** when a group is processed by a service, it expands the group (and any nested groups) into a set of addresses. The resulting set of addresses contains no groups, and any duplicate addresses are removed. Thus, a resolved group may be considered an exclusive union of all of its contained members

**network managed group:** group created and managed within a network, allowing Web Services to reference the members of a group using the group name

### 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 202 391-1 [2] apply.

---

## 4 Detailed service description

The present document defines two related interfaces, one to manage the groups themselves - creation, deletion, query and access right management. The second interface manages the members within a group, supporting add, delete and query operations.

Addresses are not created using this service, they must already exist.

### 4.1 Group URI format

A group URI is consistent with the style defined in RFC 2396 [3], supporting the following URI style which is used in schemes such as sip and mailto:

scheme:dept1294@mydivision.mycompany.serviceprovider.com

The group URI consists of the following discrete elements:

**Scheme:** selected by the provider of the group URI.

**Group name:** following the conventions of RFC 2396 [3].

**Suffix:** may be added by Service Provider (if allowed by creation operation) to create a unique name when the Prefix + Group name already exists.

**Sub-domain:** defined by the requester, this is contained within the domain provided by the service provider.

**Domain:** defined by the Service Provider, and cannot be specified by the application.

This definition of a group URI enables flexibility on the part of the Service Provider and the Requester, while ensuring unique groups are created and providing transparency of implementation of group storage.

The following are some group URI examples.

- sip:salesteam@sales.acme.anytelco.com
- sip:salesteam1@sales.acme.anytelco.com
- mailto:fieldservice@cityofaustin.anytelco.com
- group:mailroom@bldg001.acme.anytelco.com

These examples show (1)(2) use of prefix to create unique names, (1)(3) use of different defined schemes, and (4) use of a service provider defined scheme.

## 4.2 Address list usage in services

When a service has a requirement to support groups of address lists, it may satisfy this requirement by utilizing network managed groups. The group URI is passed to the service, and this group URI is resolved to the set of URIs contained within the group. If one or more group URIs are provided in a set of URIs to a service, the service will replace each group URI with its set of contained URIs, and the service processing will apply to the unique union of URIs generated.

If supported by the service policy, zero or more of the set of URIs contained within a group may be themselves group URIs, which would also be resolved. Thus, in this case, the list of URIs that the service would process would be the union of individual URIs (as a set with no duplicates).

Unless specifically defined in the semantics of a service, the expected semantic for the results of a service operation will be presented as the results for the set of URIs as processed (the union of non-group and group provided URIs), without group URIs included in the result. This eliminates a variety of complexity issues including duplicate URIs in multiple groups and the differences between a group URI and a URI referring to an endpoint.

---

# 5 Namespaces

The GroupManagement interface uses the namespace:

`http://www.csapi.org/wsdl/parlayx/group_mgmt/v2_2`

The Group interface uses the namespace:

`http://www.csapi.org/wsdl/parlayx/group/v2_2`

The GroupMember interface uses the namespace:

`http://www.csapi.org/wsdl/parlayx/group_member/v2_2`

The data types are defined in the namespace:

`http://www.csapi.org/schema/parlayx/group/v2_1`

The "xsd" namespace is used in the present document to refer to the XML Schema data types defined in XML Schema [1]. The use of the name "xsd" is not semantically significant.



## 6 Sequence diagrams

### 6.1 Manage groups (Create, delete, query, set access and query access)

Pattern: Request / Response.

The group management functions are shown in this diagram, showing a sequence including the creation of a group, setting access permissions to the group, querying those permissions, query of groups and finally deletion of a group.

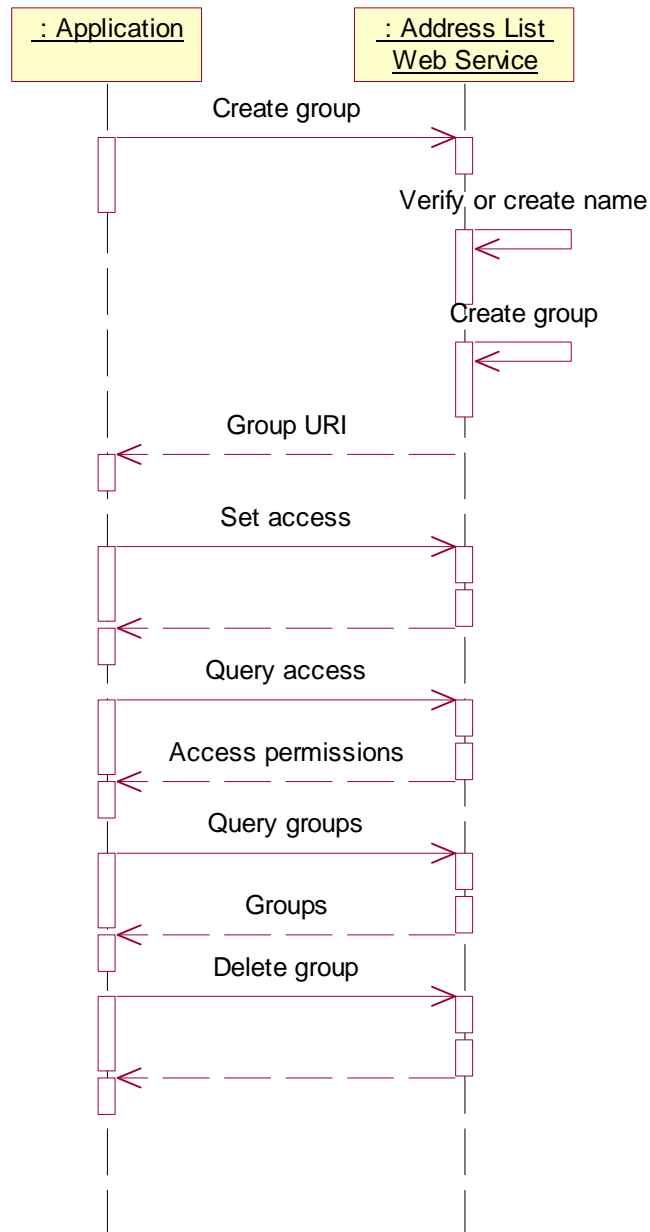


Figure 1

## 6.2 Manage group members (AddMember, AddMembers, DeleteMember, DeleteMembers, QueryMembers)

Pattern: Request / Response.

The group membership functions are shown in this diagram, showing the two add, two delete, and the query function.

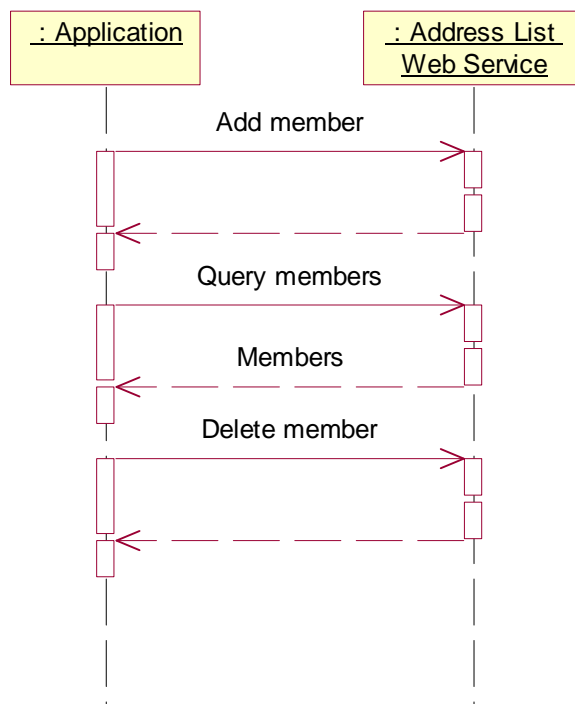


Figure 2

## 7 XML Schema data type definition

### 7.1 AccessPermissions structure

List of access permissions that may be assigned to a requester associated with a group.

Element name	Element type	Optional	Description
adminPermission	xsd:boolean	No	Requester has admin permission for the group
addPermission	xsd:boolean	No	Requester can add members to a group
deletePermission	xsd:boolean	No	Requester can delete members from a group
queryPermission	xsd:boolean	No	Requester can query members in a group

### 7.2 AttributeStatus enumeration

Enumeration value	Description
Valid	Attribute is valid
Unknown	Attribute is not defined
Denied	Access to the attribute is denied

## 7.3 SimpleAttribute structure

Attribute representing a name and an associated value.

Element name	Element type	Optional	Description
name	xsd:string	No	Name of the attribute
type	xsd:string	No	Type of the attribute. The value is always a string, but this provides information on the format of the value
value	xsd:string	No	Value of the attribute
status	AttributeStatus	No	Status of the attribute

## 8 Web Service interface definition

The Address List Management service consists of three interfaces:

- GroupManagement which manages creation and access to groups that hold the address lists.
- Group which manages the content of the address list.
- GroupMember which represents an address list entry and its associated properties.

Together these provide the interfaces to create and manage address lists, enabling these groups to be used by other services through this common capability.

### 8.1 Interface: GroupManagement

The GroupManagement interface provides the administration interface for creating, deleting, querying and managing access rights for groups. The format of the group name is specified in the Detailed Service Description (see clause 4).

#### 8.1.1 Operation: createGroup

Create a new group. The requester provides the name for the group and the domain segment in which the group is to be stored. A domain segment is used, since the full domain will consist of the domain segment provided by the requester (e.g. "sales.mycompany") plus a period separator (".") per RFC 2396 [3] and the domain segment provided by the Service Provider (e.g. "serviceprovider.com").

To avoid name conflicts, since group URIs must be unique, an automatic naming capability is provided which will append a suffix to the name provided if the name is already used within the domain. If the AutoName is set to "true" and the fully qualified name is not unique, then the name will have a suffix added and the unique name will be provided in the result. For example, if the group "sales@mycompany.serviceprovider.com" was already defined, a suffix would be added and the result could be "sales1@mycompany.serviceprovider.com". If the AutoName is set to "false", then a PolicyException is thrown if the group URI is not unique.

##### 8.1.1.1 Input message: createGroupRequest

Part name	Part type	Optional	Description
name	xsd:string	No	Name of group to be included in group name
domain	xsd:string	No	Domain segment to be contained within the domain provided by the Service Provider. May be hierarchical using period separators (see RFC 2396 [3])
autoName	xsd:boolean	No	If false, name must be unique or it will not be created. If true, a suffix will be added to the name if it is not unique

### 8.1.1.2 Output message: createGroupResponse

Part name	Part type	Optional	Description
result	xsd:anyURI	No	Fully qualified group name

### 8.1.1.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.
- POL0212: Group name too long.
- POL0213: Group already exists.

## 8.1.2 Operation: deleteGroup

Delete a group.

### 8.1.2.1 Input message: deleteGroupRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Name of group to delete

### 8.1.2.2 Output message: deleteGroupResponse

Part name	Part type	Optional	Description
None			

### 8.1.2.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.1.3 Operation: queryGroups

Group information can be retrieved from the network, with two types of search, one that retrieves groups only from a single sub-domain and one that returns groups from the sub-domain and its sub-domains.

An example demonstrates the two search types. The following example data is used:

- Dept123@region1.sales.mycompany.serviceprovider.com
- Dept245@region2.sales.mycompany.serviceprovider.com
- Dept348@sales.mycompany.serviceprovider.com

- Dept367@sales.mycompany.serviceprovider.com
- Dept875@finance.mycompany.serviceprovider.com

For a search using the search domain "sales.mycompany", with the hierarchy set to "false", the result will contain:

- Dept348@sales.mycompany.serviceprovider.com
- Dept367@sales.mycompany.serviceprovider.com

If the same search domain "sales.mycompany" is used, but the hierarchy set to "true", the result will contain:

- Dept123@region1.sales.mycompany.serviceprovider.com
- Dept245@region2.sales.mycompany.serviceprovider.com
- Dept348@sales.mycompany.serviceprovider.com
- Dept367@sales.mycompany.serviceprovider.com

#### 8.1.3.1 Input message: queryGroupsRequest

Part name	Part type	Optional	Description
searchDomain	xsd:string	No	Sub-domain to retrieve groups from
hierarchy	xsd:boolean	No	Follow hierarchy under search name

#### 8.1.3.2 Output message: queryGroupsResponse

Part name	Part type	Optional	Description
result	xsd:anyURI [0..unbounded]	Yes	Array of items matching search criteria

#### 8.1.3.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

#### 8.1.4 Operation: setAccess

Access to manage the elements within a group may be provided independently from the access to manage the group itself. This operation enables the group administrator to specify the requester and the operations the requester is permitted to perform through the Group interface.

The access rights are absolute, if a requester has "query" access currently and "add" access is to be added, then the request requires both "add" and "query" rights to be set to "true". Likewise, any right that is set to "false" will be revoked.

### 8.1.4.1 Input message: setAccessRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to grant access to
requester	xsd:string	No	Requester to grant access to
adminPermission	xsd:Boolean	No	Permission to manage group
addPermission	xsd:Boolean	No	Permission to add members to the group
deletePermission	xsd:Boolean	No	Permission to delete members from the group
queryPermission	xsd:Boolean	No	Permission to query members in the group

### 8.1.4.2 Output message: setAccessResponse

Part name	Part type	Optional	Description
None			

### 8.1.4.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.1.5 Operation: queryAccess

Query the access permissions for a requester on a group.

### 8.1.5.1 Input message: queryAccessRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to which permissions are to be granted
requester	xsd:string	No	Requester to retrieve access permissions for

### 8.1.5.2 Output message: queryAccessResponse

Part name	Part type	Optional	Description
result	AccessPermissions	No	List of permissions that a requester has

### 8.1.5.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2 Interface: Group

The Group interface provides the administration interface for creating, deleting, querying members within a group.

### 8.2.1 Operation: addMember

Add a member to a group. If the new member is a group, and if nested group support is provided, this will add the group URI as a reference to the list of members (it will not expand the contents of the group within this group). A group may not be added recursively, an attempt to do so will result in a ServiceException.

To add a group as a member of a group, the requester must have query permission on the group to be added.

#### 8.2.1.1 Input message: addMemberRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	URI of group
member	xsd:anyURI	No	Member to add to the group

#### 8.2.1.2 Output message: addMemberResponse

Part name	Part type	Optional	Description
None			

#### 8.2.1.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.
- POL0210: Too many members in group.
- POL0211: Subgroups not allowed.

### 8.2.2 Operation: addMembers

Add an array of members to a group. If nested group support is provided, this will add any group URIs, as references, to the list of members (it will not expand the contents of any groups within this group). No group may be added recursively, an attempt to do so will result in a ServiceException, and none of the members will be added to the group.

To add a group as a member of a group, the requester must have query permission on the group to be added.

#### 8.2.2.1 Input message: addMembersRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	URI of group
members	xsd:anyURI [1..unbounded]	No	Member(s) to add to the group

#### 8.2.2.2 Output message: addMembersResponse

Part name	Part type	Optional	Description
None			

### 8.2.2.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.
- POL0210: Too many members in group.
- POL0211: Subgroups not allowed.

## 8.2.3 Operation: deleteMember

Delete a member from a group. The member may only be removed from this group. If nested groups are supported, the member will not be removed from any nested group. Removal of a group URI will remove that group URI reference from this group, is will not delete the group.

### 8.2.3.1 Input message: deleteMemberRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	URI of group
member	xsd:anyURI	No	Member to delete from the group

### 8.2.3.2 Output message: deleteMemberResponse

Part name	Part type	Optional	Description
None			

### 8.2.3.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.4 Operation: deleteMembers

Delete an array of members from a group. The members may only be removed from this group. If nested groups are supported, the members will not be removed from any nested group. Removal of a group URI will remove that group URI reference from this group, it will not delete the group. If the array contains URIs that are not in the group, they will be ignored and no fault will be generated.

### 8.2.4.1 Input message: deleteMembersRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	URI of group
members	xsd:anyURI [1..unbounded]	No	Member(s) to delete from the group



### 8.2.4.2 Output message: deleteMembersResponse

Part name	Part type	Optional	Description
None			

### 8.2.4.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.5 Operation: queryMembers

Get the list of members contained within a group.

If nested groups are supported, then the member list may contain group URIs as members. Therefore, two manners are supported for retrieving the list of members - with members resolved and without.

- If **resolveGroups** is "true", then the exclusive union of all the members contained within the group, and any nested subgroups, is the result (exclusive union means that after retrieving all members, duplicate members are removed).
- If **resolveGroups** is "false", then the group members are returned including group URIs as members of the group. If members within nested groups are required, subsequent calls to this operation with those groups may be used to retrieve those members.

If nested groups are not supported, the value of **resolveGroups** is ignored.

### 8.2.5.1 Input message: queryMembersRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	URI of group
resolveGroups	xsd:boolean	No	If true, return set of members after resolving groups (including subgroups). If false, return members including group references

### 8.2.5.2 Output message: queryMembersResponse

Part name	Part type	Optional	Description
result	xsd:anyURI [0..unbounded]	Yes	Members of group

### 8.2.5.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.6 Operation: addGroupAttribute

Groups may have attributes associated with the group. To avoid conflicts, attribute names that start with Group are reserved for use as defined within the present document:

- Group.Description.
- Group.ExpiryDate.

Attributes may be added or updated by those with admin or add permission on the specified group.

### 8.2.6.1 Input message: addGroupAttributeRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to set attribute for
value	SimpleAttribute	No	Attribute to add, or update

### 8.2.6.2 Output message: addGroupAttributeResponse

Part name	Part type	Optional	Description
None			

### 8.2.6.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.7 Operation: deleteGroupAttribute

Groups may have attributes removed by those with admin or delete permission on the specified group.

### 8.2.7.1 Input message: deleteGroupAttributeRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to delete attribute from
attributeName	xsd:string	No	Name of attribute to delete

### 8.2.7.2 Output message: deleteGroupAttributeResponse

Part name	Part type	Optional	Description
None			

### 8.2.7.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.8 Operation: queryGroupAttributes

Query the attributes for a group by those with admin or read permission on the specified group.

### 8.2.8.1 Input message: queryGroupAttributesRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to get attributes for

### 8.2.8.2 Output message: queryGroupAttributesResponse

Part name	Part type	Optional	Description
result	SimpleAttribute [0..unbounded]	Yes	Group attributes

### 8.2.8.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.9 Operation: addGroupMemberAttribute

Group members may have attributes that are within the context of a group in which they belong.

Group member attributes may be added or updated by those with admin or add permission on the specified group.

### 8.2.9.1 Input message: addGroupMemberAttributeRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to set attribute for
member	xsd:anyURI	No	Member to set attribute for
value	SimpleAttribute	No	Attribute to add, or update

### 8.2.9.2 Output message: addGroupMemberAttributeResponse

Part name	Part type	Optional	Description
None			

### 8.2.9.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.10 Operation: deleteGroupMemberAttribute

Group members may have attributes removed by those with admin or delete permission on the specified group.

### 8.2.10.1 Input message: deleteGroupMemberAttributeRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to delete attribute from
member	xsd:anyURI	No	Member to delete attribute from
attributeName	xsd:string	No	Name of attribute to remove

### 8.2.10.2 Output message: deleteGroupMemberAttributeResponse

Part name	Part type	Optional	Description
None			

### 8.2.10.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.2.11 Operation: queryGroupMemberAttributes

Query the attributes for a group member by those with admin or read permission on the specified group.

### 8.2.11.1 Input message: queryGroupMemberAttributesRequest

Part name	Part type	Optional	Description
group	xsd:anyURI	No	Group to get attributes for
member	xsd:anyURI	No	Member to get attributes for

### 8.2.11.2 Output message: queryGroupMemberAttributesResponse

Part name	Part type	Optional	Description
result	SimpleAttribute [0..unbounded]	Yes	Group member attributes

### 8.2.11.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.3 Interface: Member

The Member interface provides access to information related to a particular entity.

### 8.3.1 Operation: addMemberAttribute

Add member attribute. If an attribute with this name exists, its value will be replaced with the value provided in this operation.

#### 8.3.1.1 Input message: addMemberAttributeRequest

Part name	Part type	Optional	Description
member	xsd:anyURI	No	Member to add attribute to
data	SimpleAttribute	No	Attribute to add to member

#### 8.3.1.2 Output message: addMemberAttributeResponse

Part name	Part type	Optional	Description
None			

#### 8.3.1.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

### 8.3.2 Operation: queryMemberAttributes

Query attributes of a member. If any attributes requested do not exist, they will not be included in the result.

#### 8.3.2.1 Input message: queryMemberAttributesRequest

Part name	Part type	Optional	Description
member	xsd:anyURI	No	Member to query attributes for
attributeNames	xsd:string [1..unbounded]	No	List of attribute names to retrieve

#### 8.3.2.2 Output message: queryMemberAttributesResponse

Part name	Part type	Optional	Description
result	SimpleAttribute [0..unbounded]	Yes	List of attributes

### 8.3.2.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 8.3.3 Operation: deleteMemberAttribute

Delete attribute from a member. If the attribute specified does not exist, it will be ignored.

### 8.3.3.1 Input message: deleteMemberAttributeRequest

Part name	Part type	Optional	Description
member	xsd:anyURI	No	Member to remove attributes from
attributeName	xsd:string	No	Name of attribute to delete

### 8.3.3.2 Output message: deleteMemberAttributeResponse

Part name	Part type	Optional	Description
None			

### 8.3.3.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.

## 9 Fault definitions

### 9.1 PolicyException

#### 9.1.1 POL0210: Too many members in group

Number of members in a group exceeds the number allowed by the Service Policy (MaxGroupMembers).

Name	Description
messageId	POL0210
text	Attempt to exceed maximum number of members in a group. Maximum number allowed is %1
variables	%1 = Maximum number allowed by Service Policy

### 9.1.2 POL0211: Subgroups not supported

Attempt to add a subgroup not permitted by Service Policy (SupportNestedGroups).

Name	Description
messageld	POL0211
text	Attempted to add a group to an existing group. Subgroups are not supported
variables	None

### 9.1.3 POL0212: Group name too long

Length of group name exceeds the length allowed by the Service Policy (MaxGroupLength).

Name	Description
messageld	POL0212
text	Group name is too long. Maximum length allowed is %1
variables	%1 = Maximum length allowed by Service Policy

### 9.1.4 POL0213: Group already exists

If the group name is not unique and the **autoName** part value is set to "false", then a PolicyException is returned since the group name already exists.

Name	Description
messageld	POL0213
text	Group URI %1 already exists. Group not created
variables	%1 = Group URI

---

## 10 Service policies

Service policies for this service.

Name	Type	Description
MaxGroupLength	xsd:int	Maximum length of the group name (user portion)
MaxGroupMembers	xsd:int	Maximum number of members in a group
SupportNestedGroups	xsd:boolean	Can a group member be a group URI

---

## Annex A (normative): WSDL for Address List Management

The document/literal WSDL representation of this interface specification is compliant to ES 202 391-1 [2] and is contained in text files (contained in archive es\_20239113v010301p0.zip) which accompany the present document.



---

## Annex B (informative): Bibliography

ETSI TR 121 905: "Universal Mobile Telecommunications System (UMTS); Vocabulary for 3GPP Specifications (3GPP TR 21.905)".

---

## History

<b>Document history</b>		
V1.1.1	March 2005	Publication
V1.2.1	December 2006	Publication
V1.3.1	February 2008	Membership Approval Procedure    MV 20080425: 2008-02-26 to 2008-04-25
V1.3.1	May 2008	Publication