

**Open Service Access (OSA);  
Parlay X Web Services;  
Part 8: Terminal Status  
(Parlay X 2)**



---

Reference

RES/TISPAN-01033-08-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 2006.

© The Parlay Group 2006.

All rights reserved.

**DECT™**, **PLUGTESTS™** and **UMTS™** are Trade Marks of ETSI registered for the benefit of its Members.  
**TIPHON™** and the **TIPHON logo** are Trade Marks currently being registered by ETSI 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  |
| 3 Definitions and abbreviations.....                        | 6  |
| 3.1 Definitions .....                                       | 6  |
| 3.2 Abbreviations .....                                     | 6  |
| 4 Detailed service description .....                        | 7  |
| 5 Namespaces.....   | 7  |
| 6 Sequence diagrams .....                                   | 8  |
| 6.1 Terminal status query .....                             | 8  |
| 6.2 Terminal status group query .....                       | 9  |
| 6.3 Terminal status notification .....                      | 10 |
| 6.4 Terminal Status Notification with Check Immediate ..... | 11 |
| 7 XML Schema data type definition .....                     | 12 |
| 7.1 Status enumeration .....                                | 12 |
| 7.2 RetrievalStatus enumeration.....                        | 12 |
| 7.3 StatusData structure.....                               | 13 |
| 7.4 StatusInformation structure .....                       | 13 |
| 8 Web Service interface definition.....                     | 13 |
| 8.1 Interface: TerminalStatus .....                         | 13 |
| 8.1.1 Operation: getStatus.....                             | 13 |
| 8.1.1.1 Input message: getStatusRequest .....               | 13 |
| 8.1.1.2 Output message: getStatusResponse .....             | 13 |
| 8.1.1.3 Referenced faults.....                              | 13 |
| 8.1.2 Operation: getStatusForGroup.....                     | 14 |
| 8.1.2.1 Input message: getStatusForGroupRequest.....        | 14 |
| 8.1.2.2 Output message: getStatusForGroupResponse.....      | 14 |
| 8.1.2.3 Referenced faults.....                              | 14 |
| 8.2 Interface: TerminalStatusNotificationManager .....      | 14 |
| 8.2.1 Operation: startNotification .....                    | 15 |
| 8.2.1.1 Input message: startNotificationRequest.....        | 15 |
| 8.2.1.2 Output message: startNotificationResponse.....      | 15 |
| 8.2.1.3 Referenced faults.....                              | 15 |
| 8.2.2 Operation: endNotification .....                      | 16 |
| 8.2.2.1 Input message: endNotificationRequest.....          | 16 |
| 8.2.2.2 Output message: endNotificationResponse.....        | 16 |
| 8.2.2.3 Referenced faults.....                              | 16 |
| 8.3 Interface: TerminalNotification.....                    | 16 |
| 8.3.1 Operation: statusNotification .....                   | 16 |
| 8.3.1.1 Input message: statusNotificationRequest.....       | 17 |
| 8.3.1.2 Output message: statusNotificationResponse .....    | 17 |
| 8.3.1.3 Referenced faults.....                              | 17 |
| 8.3.2 Operation: statusError.....                           | 17 |
| 8.3.2.1 Input message: statusErrorRequest .....             | 17 |
| 8.3.2.2 Output message: statusErrorResponse .....           | 17 |
| 8.3.2.3 Referenced faults.....                              | 17 |
| 8.3.3 Operation: statusEnd.....                             | 17 |
| 8.3.3.1 Input message: statusEndRequest .....               | 17 |
| 8.3.3.2 Output message: statusEndResponse .....             | 17 |
| 8.3.3.3 Referenced faults.....                              | 18 |

|                               |  |           |
|-------------------------------|--|-----------|
| 9                             | Fault definitions.....                     | 18        |
| 9.1                           | PolicyException .....                      | 18        |
| 9.1.1                         | POL0200: Busy criteria not supported ..... | 18        |
| 10                            | Service policies .....                     | 18        |
| <b>Annex A (normative):</b>   | <b>WSDL for Terminal Status .....</b>      | <b>19</b> |
| <b>Annex B (informative):</b> | <b>Bibliography .....</b>                  | <b>20</b> |
| History .....                 |  | 21        |

---

# 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), and is now submitted for the ETSI standards Membership Approval Procedure.

The present document is part 8 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 2.1 set of specifications.**

**The present document is equivalent to 3GPP TS 29.199-08 V6.2.0 (Release 6).**

---

# 1 Scope

The present document is part 8 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 Terminal Status 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

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- 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.
- For a non-specific reference, the latest version applies.

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

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

[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 Definitions and abbreviations

## 3.1 Definitions

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

## 3.2 Abbreviations

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

---

## 4 Detailed service description

Terminal Status provides access to the status of a terminal through:

- Request for the status of a terminal.
- Request for the status of a group of terminals.
- Notification of a change in the status of a terminal.

The status of a terminal can be expressed as reachable, unreachable or busy - however not all terminals distinguish a busy status, so applications should be able to adapt to what information is available (using the service properties to determine available information).

When a request for a group of terminals is made, the response may contain a full or partial set of results. This allows the service to provide results based on a number of criteria including number of terminals for which the request is made and amount of time required to retrieve the information. This allows the requester to initiate additional requests for those terminals for which information was not provided.

---

## 5 Namespaces

The TerminalStatus interface uses the namespace:

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

The TerminalStatusNotificationManager interface uses the namespace:

`http://www.csapi.org/wsdl/parlayx/terminal_status/notification_manager/v2_2`

The TerminalStatusNotification interface uses the namespace:

`http://www.csapi.org/wsdl/parlayx/terminal_status/notification/v2_2`

The data types are defined in the namespace:

`http://www.csapi.org/schema/parlayx/terminal_status/v2_2`

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 Terminal status query

Pattern: Request / Response.

When an application is interested in determining the status of a terminal device, it may provide a terminal device address, and receive the status for the device requested.

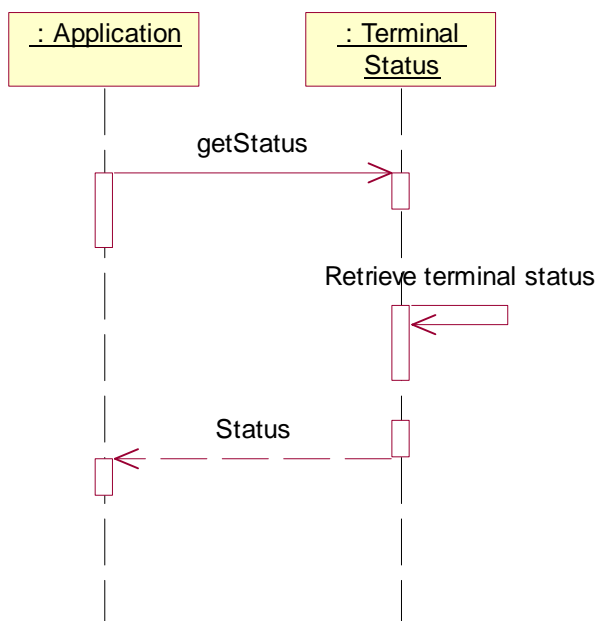


Figure 1



## 6.2 Terminal status group query

Pattern: Request / Response.

When an application is interested in determining the status of a set of terminal devices, it may provide an array of terminal device addresses, including network managed group addresses, and receive the status for the set of devices requested.

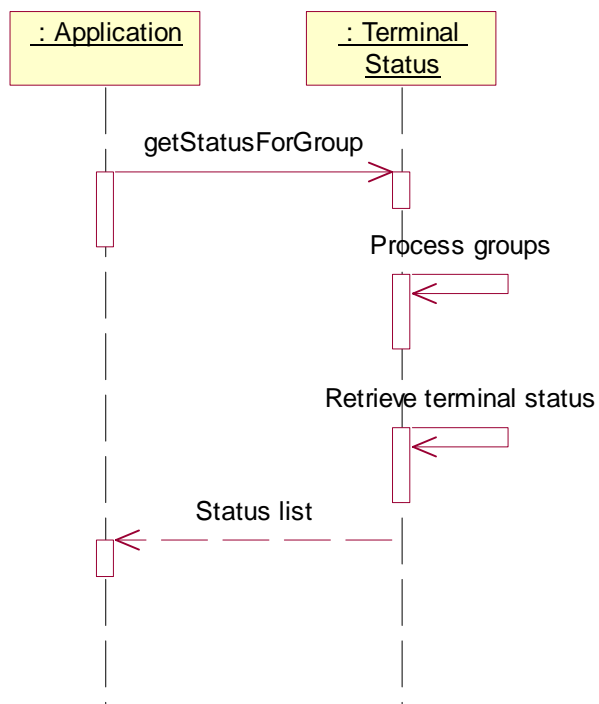


Figure 2

## 6.3 Terminal status notification

Pattern: Application Correlated Multiple Notification.

An application can be notified of a change in the status of terminal devices. When the status of a terminal device changes, a notification message will be sent to the application.

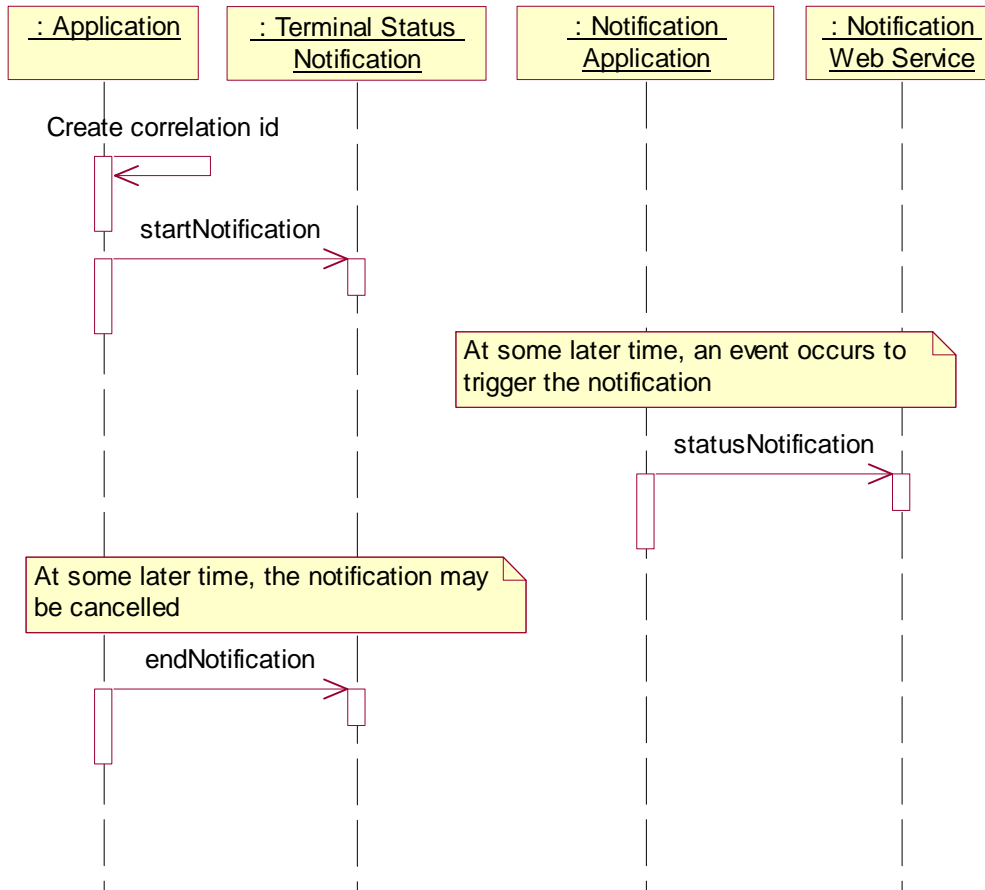


Figure 3

## 6.4 Terminal Status Notification with Check Immediate

In some applications, the terminal status notification will be used to watch for a specific status change. An example is a "call when available" service, where the terminal status is checked and determined to be not reachable or busy, and a notification is set up to notify the application when the terminal becomes reachable. Between the time that the original status determination and the time the notification is set up, the terminal status could change to reachable, thus the notification on change to reachable would not be sent.

Using the check immediate flag, after the notification is established, the value of the terminal status will be determined, and if the criteria is matched then a notification will be sent immediately. The following sequence diagram shows this scenario.

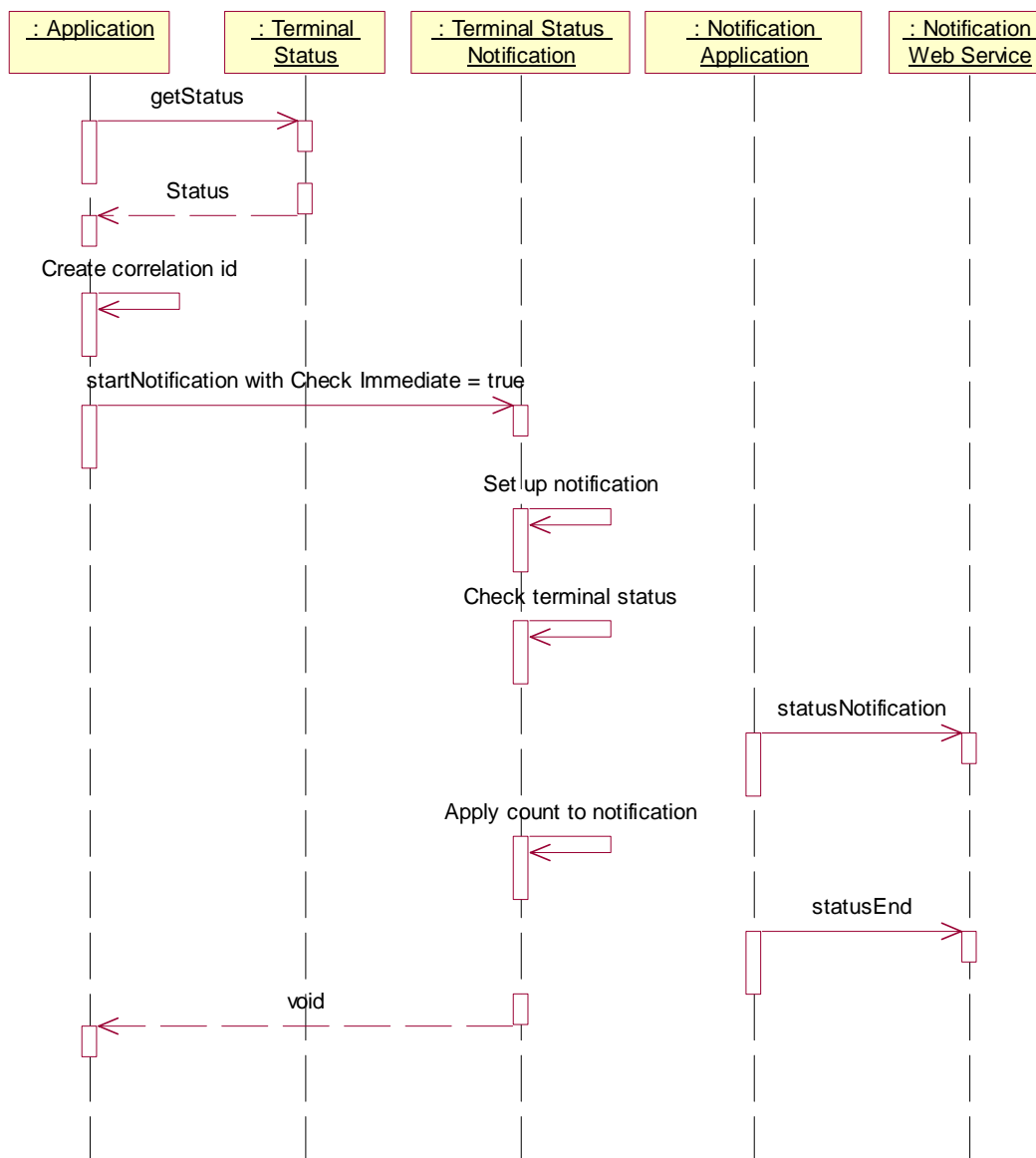


Figure 4

This sequence shows:

- The Enterprise Application checks the status of a terminal, and receives its status (in this scenario receiving Unreachable or Busy).
- The Enterprise Application generates a correlator, and starts a notification with criteria defined to notify the Enterprise Web Service when the terminal state becomes Reachable and the check immediate flag set to true.
- Sets up the notification to monitor terminal status changes.
- Check the current status of the terminal, and determine if the status matches the criteria.
- In this case, the criteria matches, and a notification is delivered to the Enterprise Web Service.
- The count of notifications is incremented and compared to the notification count limit.
- In this case, a single notification was requested, and the end notification message is sent.
- The startNotification operation completes.

This scenario includes the full set of interactions in one sequence, which also shows that the notifications can be received concurrent with the creation of the notification.

## 7 XML Schema data type definition

### 7.1 Status enumeration

List of possible status values.

| Enumeration value | Description               |
|-------------------|---------------------------|
| Reachable         | Terminal is reachable     |
| Unreachable       | Terminal is not reachable |
| Busy              | Terminal is busy          |

### 7.2 RetrievalStatus enumeration

Enumeration of the status items that are related to an individual retrieval in a set.

| Enumeration value | Description  |
|-------------------|--|
| Retrieved         | Status retrieved, with result in <b>currentStatus</b> element  |
| NotRetrieved      | Status not retrieved, <b>currentStatus</b> is not provided (does not indicate an error, no attempt may have been made) |
| Error             | Error retrieving status  |

## 7.3 StatusData structure

Data structure containing device identifier and its status. As this can be related to a query of a group of terminal devices, the **reportStatus** element is used to indicate whether the information for the device was retrieved or not, or if an error occurred.

| Element name     | Element type        | Optional | Description   |
|------------------|---------------------|----------|---|
| address          | xsd:anyURI          | No       | Address of the Terminal Device to which the status information applies  |
| reportStatus     | RetrievalStatus     | No       | Status of retrieval for this address  |
| currentStatus    | Status              | Yes      | Status of terminal. It is only provided if <b>reportStatus=Retrieved</b> .  |
| errorInformation | common:ServiceError | Yes      | If <b>reportStatus</b> is Error, this is the reason for the error. Error due to privacy verification will be expressed as POL0002 in the ServiceError |

## 7.4 StatusInformation structure

A simplified terminal status data structure used in the **TerminalNotification** interface.

| Name          | Type       | Optional | Description   |
|---------------|------------|----------|---|
| address       | xsd:anyURI | No       | Address of the terminal device to which the status information applies. |
| currentStatus | Status     | No       | Status of terminal.   |

---

# 8 Web Service interface definition

## 8.1 Interface: TerminalStatus

Request the status for a terminal or set of terminals.

### 8.1.1 Operation: getStatus

This operation is intended to retrieve the status for a single terminal. The URI provided is for a single terminal, not a group URI. If a group URI is provided, a PolicyException will be returned to the application.

#### 8.1.1.1 Input message: getStatusRequest

| Part name | Part type  | Optional | Description                    |
|-----------|------------|----------|--------------------------------|
| address   | xsd:anyURI | No       | Terminal to request status for |

#### 8.1.1.2 Output message: getStatusResponse

| Part name | Part type | Optional | Description  |
|-----------|-----------|----------|--|
| result    | Status    | No       | Status for the terminal for which status was requested |

#### 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.
- POL0002: Privacy error.
- POL0006: Groups not allowed.

## 8.1.2 Operation: getStatusForGroup

This operation initiates a retrieval activity, where one or more terminals, or groups of terminals, may have their status determined.

The Web Service may return a result set that does not include complete information, allowing the Web Service implementation to choose to deliver a partial set of results to accommodate other conditions, such as avoiding timeouts. In this case, the addresses for which no attempt was made to provide data will be marked **NotRetrieved** in the result for each address this applies to.

### 8.1.2.1 Input message: getStatusForGroupRequest

| Part name | Part type                    | Optional | Description  |
|-----------|------------------------------|----------|--|
| addresses | xsd:anyURI<br>[1..unbounded] | No       | List of URIs to get status for, including group URIs |

### 8.1.2.2 Output message: getStatusForGroupResponse

| Part name | Part type                    | Optional | Description                    |
|-----------|------------------------------|----------|--------------------------------|
| result    | StatusData<br>[1..unbounded] | No       | Set of results for the request |

### 8.1.2.3 Referenced faults

ServiceException from ES 202 391-1 [2]:

- SVC0001: Service error.
- SVC0002: Invalid input value.
- SVC0004: No valid addresses.
- SVC0006: Invalid group.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.
- POL0003: Too many addresses.
- POL0006: Groups not allowed.
- POL0007: Nested groups not allowed.

## 8.2 Interface: TerminalStatusNotificationManager

Set up notifications for terminal status changes.

## 8.2.1 Operation: startNotification

Notifications of status changes are made available to applications. The number and duration of notifications may be requested as part of the setup of the notification or may be governed by service policies, or a combination of the two.

If **checkImmediate** is set to true, then the notification will be set up, and then the current value of the terminal status will be checked. If the terminal status meets the criteria provided, a notification will be sent to the application. This notification will count against the count requested. This addresses the case where the status of the device changes during the time the notification is being set up, which may be appropriate in some applications.

The correlator provided in the reference must be unique for this Web Service at the time the notification is initiated, otherwise a ServiceException (SVC0005) will be returned to the application.

If the frequency requested is more often than allowed by the service policy, then the value in the service policy will be used. If the duration requested exceeds the time allowed in the service policy, then the value in the service policy will be used. If the notification period (duration) ends before all of the notifications (count) have been delivered, then the notification terminates. In all cases, when the notifications have run their course (by duration or count), an end of notifications message will be provided to the application.

Service policies may govern what count values can be requested, including maximum number of notifications allowed and whether unlimited notifications can be requested (i.e. either by not specifying the optional **count** message part or by specifying it with a value of zero). If the count value requested is not in policy, a PolicyException (POL0004 or POL0005 as appropriate) will be returned.

### 8.2.1.1 Input message: startNotificationRequest

| Part name      | Part type                 | Optional | Description   |
|----------------|---------------------------|----------|---|
| reference      | common:SimpleReference    | No       | Notification endpoint definition  |
| addresses      | xsd:anyURI [0..unbounded] | No       | Addresses of terminals to monitor   |
| criteria       | Status [0..unbounded]     | No       | List of status values to generate notifications for (these apply to all addresses specified)                      |
| checkImmediate | xsd:boolean               | No       | Check status immediately after establishing notification  |
| frequency      | common:TimeMetric         | No       | Maximum frequency of notifications (can also be considered minimum time between notifications)                    |
| duration       | common:TimeMetric         | Yes      | Length of time notifications occur for. Do not specify to use default notification time defined by service policy |
| count          | xsd:integer               | Yes      | Maximum number of notifications. For no maximum, either do not specify this part or specify a value of zero.      |

### 8.2.1.2 Output message: startNotificationResponse

| 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.
- SVC0004: No valid addresses.
- SVC0005: Duplicate correlator.
- SVC0006: Invalid group.

PolicyException from ES 202 391-1 [2]:

- POL0001: Policy error.
- POL0003: Too many addresses.
- POL0004: Unlimited notifications not supported.
- POL0005: Too many notifications requested.
- POL0006: Groups not allowed.
- POL0007: Nested groups not allowed.
- POL0009: Invalid frequency requested.
- POL0200: Busy criteria not supported.

## 8.2.2 Operation: endNotification

The application may end a notification using this operation. Until this operation completes, notifications may continue to be received by the application.

An end of notification (**statusEnd**) operation will not be invoked on the application for a notification ended using the **endNotification** operation.

### 8.2.2.1 Input message: endNotificationRequest

| Part name  | Part type  | Optional | Description                  |
|------------|------------|----------|------------------------------|
| correlator | xsd:string | No       | Correlator of request to end |

### 8.2.2.2 Output message: endNotificationResponse

| 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.

## 8.3 Interface: TerminalNotification

Notification interface to which notifications are delivered.

### 8.3.1 Operation: statusNotification

When the status of a monitored device changes, a notification is delivered to the application with the new status information for each of the devices. If a group identifier was used, the terminal device URI is provided, not the group URI.



### 8.3.1.1 Input message: statusNotificationRequest

| Part name      | Part type                           | Optional | Description   |
|----------------|-------------------------------------|----------|---|
| correlator     | xsd:string                          | No       | Correlator provided in request to set up this notification              |
| terminalStatus | StatusInformation<br>[1..unbounded] | No       | Set of elements, each containing a terminal address and its new status. |

### 8.3.1.2 Output message: statusNotificationResponse

| Part name | Part type | Optional | Description |
|-----------|-----------|----------|-------------|
| None      |           |          |             |

### 8.3.1.3 Referenced faults

None.

## 8.3.2 Operation: statusError

This operation is invoked on the application to indicate that the notification is being cancelled by the Web Service.

### 8.3.2.1 Input message: statusErrorRequest

| Part name  | Part type           | Optional | Description   |
|------------|---------------------|----------|---|
| correlator | xsd:string          | No       | Correlator provided in request to set up this notification.   |
| address    | xsd:anyURI          | Yes      | Address of terminal if the error applies to an individual terminal, or not specified if it applies to the whole notification. |
| reason     | common:ServiceError | No       | Reason notification is being discontinued.  |

### 8.3.2.2 Output message: statusErrorResponse

| Part name | Part type | Optional | Description |
|-----------|-----------|----------|-------------|
| None      |           |          |             |

### 8.3.2.3 Referenced faults

None.

## 8.3.3 Operation: statusEnd

The notifications have completed for this correlator. This operation will be invoked on the application when the duration or count for notifications have been completed. This operation will not be invoked in the case of an error ending the notifications or deliberate ending of the notifications (using the **endNotification** operation).

### 8.3.3.1 Input message: statusEndRequest

| Part name  | Part type  | Optional | Description   |
|------------|------------|----------|---|
| correlator | xsd:string | No       | Correlator provided in request to set up this notification. |

### 8.3.3.2 Output message: statusEndResponse

| Part name | Part type | Optional | Description |
|-----------|-----------|----------|-------------|
| None      |           |          |             |

### 8.3.3.3 Referenced faults

None.

---

## 9 Fault definitions

### 9.1 PolicyException

#### 9.1.1 POL0200: Busy criteria not supported

| Name      | Description                    |
|-----------|--------------------------------|
| messageld | POL0200                        |
| text      | Busy criteria is not supported |
| variables | None                           |

---

## 10 Service policies

Service policies for this service.

| Name                         | Type              | Description  |
|------------------------------|-------------------|--|
| BusyAvailable                | xsd:boolean       | Can busy be returned as a status or be a trigger   |
| MaximumNotificationAddresses | xsd:int           | Maximum number of addresses for which a notification can be set up   |
| MaximumNotificationFrequency | common:TimeMetric | Maximum rate of notification delivery (also can be considered minimum time between notifications)  |
| MaximumNotification Duration | common:TimeMetric | Maximum amount of time a notification may be set up for  |
| MaximumCount                 | xsd:int           | Maximum number of notifications that may be requested  |
| UnlimitedCountAllowed        | xsd:boolean       | Allowed to specify unlimited notification count (i.e. either by not specifying the optional <b>count</b> part in the <b>startNotificationRequest</b> message or by specifying a value of zero) |
| GroupSupport                 | xsd:boolean       | Groups may be included with addresses  |
| NestedGroupSupport           | xsd:boolean       | Are nested groups supported in group definitions   |

---

## Annex A (normative): WSDL for Terminal Status

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\_20239108v010201m0.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                  | October 2006 | Membership Approval Procedure    MV 20061215: 2006-10-17 to 2006-12-15 |
|                         |              |  |
|                         |              |  |
|                         |              |  |