

**Open Service Access (OSA);  
Application Programming Interface (API);  
Part 7: Terminal Capabilities SCF  
(Parlay 6)**

---



---

Reference

DES/TISPAN-01032-7-OSA

---

Keywords

API, IDL, OSA, UML

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

**3GPP**<sup>TM</sup> 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 Terminal Capabilities SCF .....	6
4.1 General requirements on support of methods.....	7
5 Sequence Diagrams .....	7
5.1 Terminal capabilities example.....	7
6 Class Diagrams.....	9
7 The Service Interface Specifications .....	10
7.1 Interface Specification Format .....	10
7.1.1 Interface Class .....	10
7.1.2 Method descriptions.....	10
7.1.3 Parameter descriptions.....	10
7.1.4 State Model.....	10
7.2 Base Interface.....	10
7.2.1 Interface Class IpInterface .....	10
7.3 Service Interfaces .....	11
7.3.1 Overview .....	11
7.4 Generic Service Interface .....	11
7.4.1 Interface Class IpService .....	11
7.4.1.1 Method setCallback().....	11
7.4.1.2 Method setCallbackWithSessionID().....	11
8 Terminal Capabilities Interface Classes .....	12
8.1 Interface Class IpTerminalCapabilities .....	12
8.1.1 Method getTerminalCapabilities().....	12
8.2 Interface Class IpExtendedTerminalCapabilities .....	13
8.2.1 Method triggeredTerminalCapabilityStartReq().....	13
8.2.2 Method triggeredTerminalCapabilityStop() .....	14
8.3 Interface Class IpAppExtendedTerminalCapabilities .....	14
8.3.1 Method triggeredTerminalCapabilityReport().....	15
8.3.2 Method triggeredTerminalCapabilityReportErr().....	15
9 State Transition Diagrams .....	16
10 Service Properties.....	16
11 Terminal Capabilities Data Definitions .....	16
11.1 terminalIdentity .....	16
11.2 TpTerminalCapabilities .....	16
11.3 TpTerminalCapabilitiesError .....	17
11.4 TpTerminalCapabilityChangeCriteria .....	17
11.5 TpTerminalCapabilityScopeType .....	17
11.6 TpTerminalCapabilityScope.....	17
12 Exception Classes.....	18
<b>Annex A (normative):      OMG IDL Description of Terminal Capabilities SCF .....</b>	<b>19</b>
<b>Annex B (informative):    W3C WSDL Description of Terminal Capabilities SCF .....</b>	<b>20</b>

<b>Annex C (informative):</b>	<b>Java™ API Description of the Terminal Capabilities SCF .....</b>	<b>21</b>
<b>Annex D (informative):</b>	<b>Contents of 3GPP OSA R7 Terminal Capabilities .....</b>	<b>22</b>
<b>Annex E (informative):</b>	<b>Description of Terminal Capabilities SCF for 3GPP2 cdma2000 networks.....</b>	<b>23</b>
E.1	General Exceptions.....	23
E.2	Specific Exceptions .....	23
E.2.1	Clause 1: Scope .....	23
E.2.2	Clause 2: References .....	23
E.2.3	Clause 3: Definitions and abbreviations .....	23
E.2.4	Clause 4: Terminal Capabilities SCF .....	23
E.2.5	Clause 5: Sequence Diagrams .....	23
E.2.6	Clause 6: Class Diagrams.....	23
E.2.7	Clause 7: The Service Interface Specifications .....	23
E.2.8	Clause 8: Terminal Capabilities Interface Classes .....	24
E.2.9	Clause 9: State Transition Diagrams .....	24
E.2.10	Clause 10: Service Properties.....	24
E.2.11	Clause 11: Terminal Capabilities Data Definitions .....	24
E.2.12	Clause 12: Exception Classes.....	24
E.2.13	Annex A (normative): OMG IDL Description of Terminal Capabilities SCF .....	24
E.2.14	Annex B (informative): W3C WSDL Description of Terminal Capabilities SCF .....	24
E.2.15	Annex C (informative): Java™ API Description of Terminal Capabilities SCF.....	24
<b>Annex F (informative):</b>	<b>Record of changes .....</b>	<b>25</b>
F.1	Interfaces .....	25
F.1.1	New .....	25
F.1.2	Deprecated.....	25
F.1.3	Removed.....	25
F.2	Methods .....	25
F.2.1	New .....	25
F.2.2	Deprecated.....	25
F.2.3	Modified.....	26
F.2.4	Removed.....	26
F.3	Data Definitions .....	26
F.3.1	New .....	26
F.3.2	Modified.....	26
F.3.3	Removed.....	26
F.4	Service Properties.....	26
F.4.1	New .....	26
F.4.2	Deprecated.....	27
F.4.3	Modified.....	27
F.4.4	Removed.....	27
F.5	Exceptions .....	27
F.5.1	New .....	27
F.5.2	Modified.....	27
F.5.3	Removed.....	27
F.6	Others .....	27
History	.....	28

---

# 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 7 of a multi-part deliverable covering Open Service Access (OSA); Application Programming Interface (API), as identified below. The API specification (ES 204 915) is structured in the following parts:

- Part 1: "Overview";
- Part 2: "Common Data Definitions";
- Part 3: "Framework";
- Part 4: "Call Control";
- Part 5: "User Interaction SCF";
- Part 6: "Mobility SCF";
- Part 7: "Terminal Capabilities SCF";**
- Part 8: "Data Session Control SCF";
- Part 9: "Generic Messaging SCF";
- Part 10: "Connectivity Manager SCF";
- Part 11: "Account Management SCF";
- Part 12: "Charging SCF";
- Part 13: "Policy Management SCF";
- Part 14: "Presence and Availability Management SCF";
- Part 15: "Multi-Media Messaging SCF";
- Part 16: "Service Broker SCF".

The present document has been defined jointly between ETSI, The Parlay Group (<http://www.parlay.org>) and the 3GPP, in co-operation with a number of JAIN™ Community (<http://www.java.sun.com/products/jain>) member companies.

**The present document forms part of the Parlay 6.0 set of specifications.**

**The present document is equivalent to 3GPP TS 29.198-7 V7.0.0 (Release 7).**

---

# 1 Scope

The present document is part 7 of the Stage 3 specification for an Application Programming Interface (API) for Open Service Access (OSA).

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

The present document specifies the Terminal Capabilities Service Capability Feature (SCF) aspects of the interface. All aspects of the Terminal Capabilities SCF are defined here, these being:

- Sequence Diagrams.
- Class Diagrams.
- Interface specification plus detailed method descriptions.
- State Transition diagrams.
- Data Definitions.
- IDL Description of the interfaces.
- WSDL Description of the interfaces.

The process by which this task is accomplished is through the use of object modelling techniques described by the Unified Modelling Language (UML).

---

# 2 References

The references listed in clause 2 of ES 204 915-1 contain provisions which, through reference in this text, constitute provisions of the present document.

ETSI ES 204 915-1: "Open Service Access (OSA); Application Programming Interface (API); Part 1: Overview (Parlay 6)".

---

# 3 Definitions and abbreviations

## 3.1 Definitions

For the purposes of the present document, the terms and definitions given in ES 204 915-1 apply.

## 3.2 Abbreviations

For the purposes of the present document, the abbreviations given in ES 204 915-1 apply.

---

# 4 Terminal Capabilities SCF

The following clauses describe each aspect of the Terminal Capabilities Capability Feature (SCF).

The order is as follows:

- The Sequence diagrams give the reader a practical idea of how each of the SCF is implemented.
- The Class relationships clause shows how each of the interfaces applicable to the SCF, relate to one another.

- The Interface specification clause describes in detail each of the interfaces shown within the Class diagram part.
- The State Transition Diagrams (STD) show the transition between states in the SCF. The states and transitions are well-defined; either methods specified in the Interface specification or events occurring in the underlying networks cause state transitions.
- The Data Definitions clause shows a detailed expansion of each of the data types associated with the methods within the classes. Note that some data types are used in other methods and classes and are therefore defined within the Common Data types part ES 204 915-2.

## 4.1 General requirements on support of methods

An implementation of this API which supports or implements a method described in the present document, shall support or implement the functionality described for that method, for at least one valid set of values for the parameters of that method.

Where a method is not supported by an implementation of a Service interface, the exception `P_METHOD_NOT_SUPPORTED` shall be returned to any call of that method.

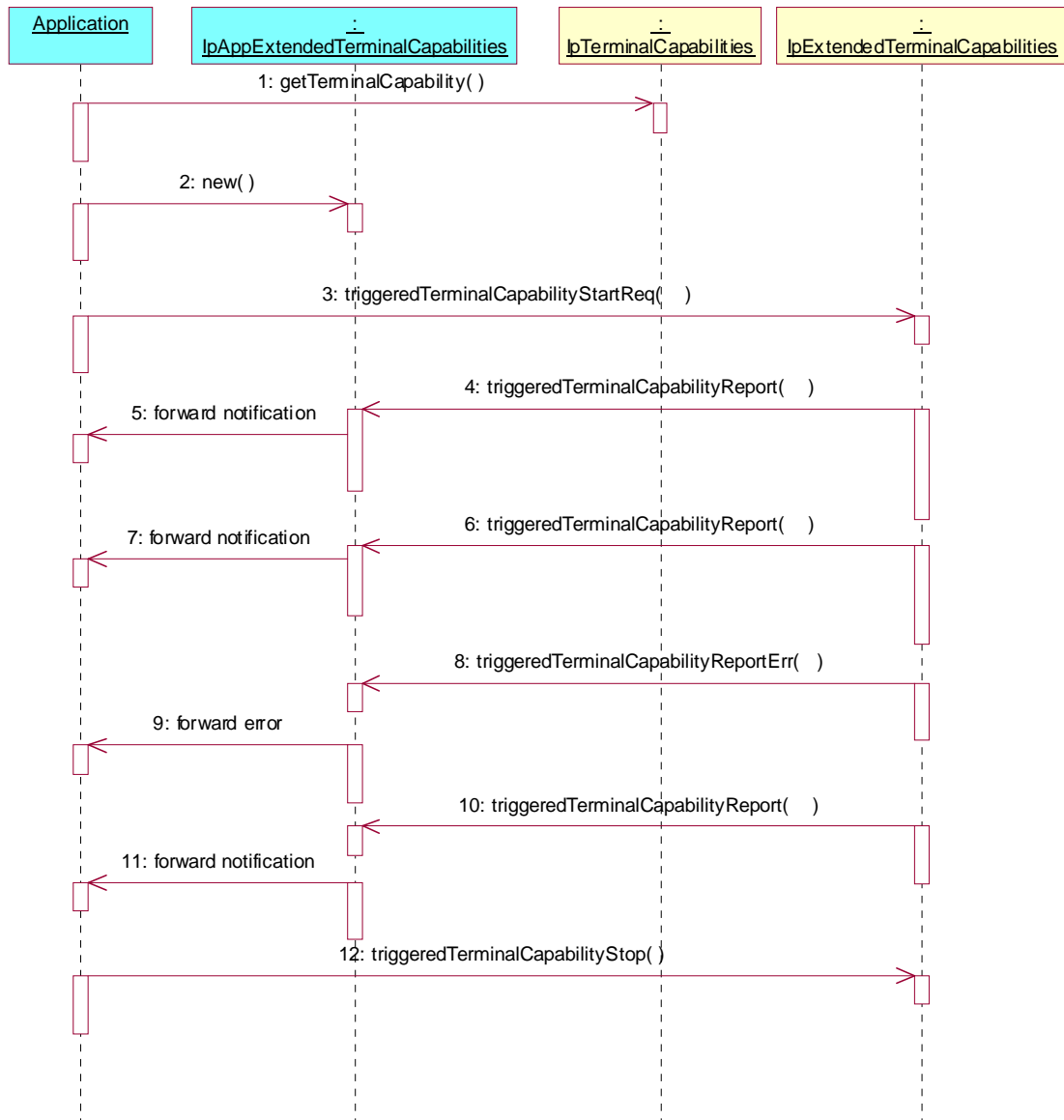
Where a method is not supported by an implementation of an Application interface, a call to that method shall be possible, and no exception shall be returned.

---

# 5 Sequence Diagrams

## 5.1 Terminal capabilities example

The following example sequence diagram illustrates how the terminal capabilities can be retrieved and their changes monitored.



- 1: The application retrieves the terminal capability of a terminal.
- 2: The application creates an object to implement IpAppExtendedTerminalCapabilities.
- 3: The terminal capabilities changes are started to be monitored.
- 4: The terminal capabilities have changed and they are reported as requested.
- 5: The report is forwarded internally to the application.
- 6: The terminal capabilities have changed and they are reported as requested.
- 7: The report is forwarded internally to the application.
- 8: An error has happened in the monitoring and it is reported.
- 9: The error report is forwarded internally to the application.
- 10: The terminal capabilities have changed and they are reported as requested.
- 11: The report is forwarded internally to the application.
- 12: The terminal capability monitoring is stopped.



## 6 Class Diagrams

Terminal Capabilities Class Diagram:

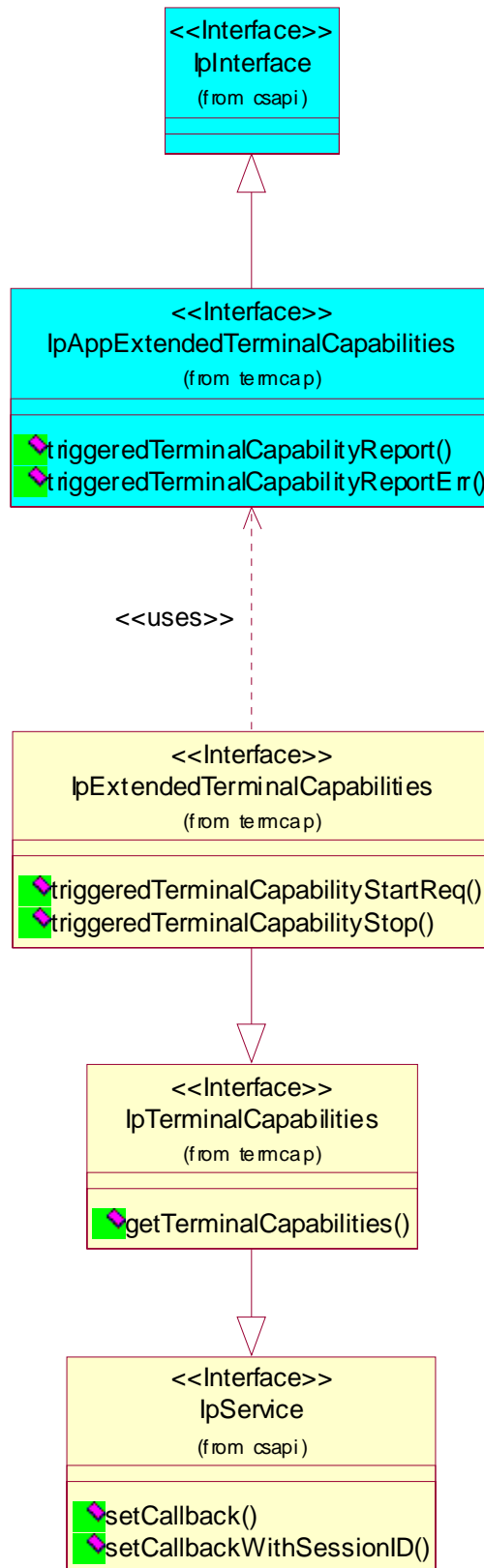


Figure 1: Terminal Capabilities Class Diagram

## 7 The Service Interface Specifications

### 7.1 Interface Specification Format

This clause defines the interfaces, methods and parameters that form a part of the API specification. The Unified Modelling Language (UML) is used to specify the interface classes. The general format of an interface specification is described below.

#### 7.1.1 Interface Class

This shows a UML interface class description of the methods supported by that interface, and the relevant parameters and types. The Service and Framework interfaces for enterprise-based client applications are denoted by classes with name Ip<name>. The callback interfaces to the applications are denoted by classes with name IpApp<name>. For the interfaces between a Service and the Framework, the Service interfaces are typically denoted by classes with name IpSvc<name>, while the Framework interfaces are denoted by classes with name IpFw<name>.

#### 7.1.2 Method descriptions

Each method (API method “call”) is described. Both synchronous and asynchronous methods are used in the API. Asynchronous methods are identified by a 'Req' suffix for a method request, and, if applicable, are served by asynchronous methods identified by either a 'Res' or 'Err' suffix for method results and errors, respectively. To handle responses and reports, the application or service developer must implement the relevant IpApp<name> or IpSvc<name> interfaces to provide the callback mechanism.

#### 7.1.3 Parameter descriptions

Each method parameter and its possible values are described. Parameters described as 'in' represent those that must have a value when the method is called. Those described as 'out' are those that contain the return result of the method when the method returns.

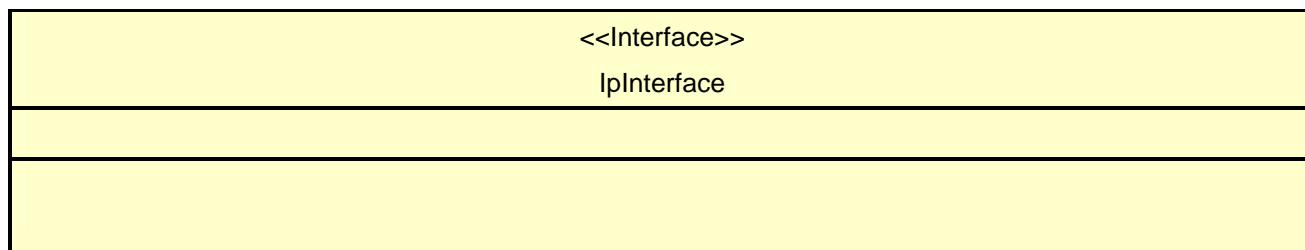
#### 7.1.4 State Model

If relevant, a state model is shown to illustrate the states of the objects that implement the described interface.

### 7.2 Base Interface

#### 7.2.1 Interface Class IpInterface

All application, framework and service interfaces inherit from the following interface. This API Base Interface does not provide any additional methods.



## 7.3 Service Interfaces

### 7.3.1 Overview

The Service Interfaces provide the interfaces into the capabilities of the underlying network - such as call control, user interaction, messaging, mobility and connectivity management.

The interfaces that are implemented by the services are denoted as 'Service Interface'. The corresponding interfaces that must be implemented by the application (e.g. for API callbacks) are denoted as 'Application Interface'.

## 7.4 Generic Service Interface

### 7.4.1 Interface Class IpService

Inherits from: IpInterface;

All service interfaces inherit from the following interface.

<<Interface>> IpService
<pre> setCallback (appInterface : in IpInterfaceRef) : void setCallbackWithSessionID (appInterface : in IpInterfaceRef, sessionID : in TpSessionID) : void </pre>

#### 7.4.1.1 Method setCallback()

This method specifies the reference address of the callback interface that a service uses to invoke methods on the application. It is not allowed to invoke this method on an interface that uses SessionIDs. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

#### *Parameters*

**appInterface : in IpInterfaceRef**

Specifies a reference to the application interface, which is used for callbacks.

#### *Raises*

**TpCommonExceptions, P\_INVALID\_INTERFACE\_TYPE**

#### 7.4.1.2 Method setCallbackWithSessionID()

This method specifies the reference address of the application's callback interface that a service uses for interactions associated with a specific session ID: e.g. a specific call, or call leg. It is not allowed to invoke this method on an interface that does not use SessionIDs. Multiple invocations of this method on an interface shall result in multiple callback references being specified. The SCS shall use the most recent callback interface provided by the application using this method. In the event that a callback reference fails or is no longer available, the next most recent callback reference available shall be used.

*Parameters***appInterface**: in IpInterfaceRef

Specifies a reference to the application interface, which is used for callbacks.

**sessionId**: in TpSessionID

Specifies the session for which the service can invoke the application's callback interface.

*Raises***TpCommonExceptions**, P\_INVALID\_SESSION\_ID, P\_INVALID\_INTERFACE\_TYPE

---

## 8 Terminal Capabilities Interface Classes

The Terminal Capabilities SCF enables the application to retrieve the terminal capabilities of the specified terminal. Additionally it is possible for the application to request notifications when the capabilities of the terminal change in some way. The Terminal Capabilities service provides SCF interfaces IpTerminalCapabilities and IpExtendedTerminalCapabilities. The application side interface for the reporting is called IpAppExtendedTerminalCapabilities.

### 8.1 Interface Class IpTerminalCapabilities

Inherits from: IpService.

The Terminal Capabilities SCF interface IpTerminalCapabilities contains the synchronous method getTerminalCapabilities. The application has to provide the terminalIdentity as input to this method. The result indicates whether or not the terminal capabilities are available in the network and, in case they are, it will return the terminal capabilities (see the data definition of TpTerminalCapabilities for more information). The network may override some capabilities that have been indicated by the terminal itself due to network policies or other restrictions or modifications in the supported capabilities.

This interface, or IpExtendedTerminalCapabilities shall be implemented by a Terminal Capabilities SCF as a minimum requirement. If this interface is implemented, the getTerminalCapabilities() method shall be implemented as a minimum requirement.

<<Interface>> IpTerminalCapabilities
getTerminalCapabilities (terminalIdentity : in TpString) : TpTerminalCapabilities

#### 8.1.1 Method getTerminalCapabilities()

This method is used by an application to get the capabilities of a user's terminal. Direction: Application to Network.

Returns result : Specifies the latest available capabilities of the user's terminal.

This information, if available, is returned as CC/PP headers as specified in W3C (see [6] in ES 204 915-1) and adopted in the WAP UAPProf specification (see [9] in ES 204 915-1). It contains URLs; terminal attributes and values, in RDF format; or a combination of both.

*Parameters***terminalIdentity**: in TpString

Identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy.

*Returns***TpTerminalCapabilities***Raises***TpCommonExceptions**, P\_INVALID\_TERMINAL\_ID

## 8.2 Interface Class IpExtendedTerminalCapabilities

Inherits from: IpTerminalCapabilities.

This interface can be used as an extended version of terminal capability monitoring. The application programmer can use this interface to request terminal capability reports that are triggered by their changes. Note that the underlying mechanisms for this network feature are currently not fully standardised.

This interface, or IpTerminalCapabilities, shall be implemented by a Terminal Capabilities SCF as a minimum requirement. The triggeredTerminalCapabilityStartReq() and triggeredTerminalCapabilityStop() methods shall be implemented as a minimum requirement. An implementation of IpExtendedTerminalCapabilities is not required to implement the minimum mandatory methods of IpTerminalCapabilities.

<<Interface>> IpExtendedTerminalCapabilities
triggeredTerminalCapabilityStartReq (appTerminalCapabilities : in IpAppExtendedTerminalCapabilitiesRef, terminals : in TpAddressSet, capabilityScope : in TpTerminalCapabilityScope, criteria : in TpTerminalCapabilityChangeCriteria) : TpAssignmentID triggeredTerminalCapabilityStop (assignmentID : in TpAssignmentID) : void

### 8.2.1 Method triggeredTerminalCapabilityStartReq()

Request for terminal capability reports when the capabilities change or when the application obviously does not have the current terminal capability information when this method is invoked.

Returns: assignmentID.

Specifies the assignment ID of the triggered terminal capability reporting request.

*Parameters***appTerminalCapabilities**: in IpAppExtendedTerminalCapabilitiesRef

Specifies the application interface for callbacks.

**terminals**: in TpAddressSet

Specifies the terminal(s) for which the capabilities shall be reported. TpAddress fields have the following use:

- Plan: Used to indicate the numbering plan.
- AddrString: Used to indicate the subscriber address.

- Name: Used to indicate the terminal identity. May be applied also together with AddrString to indicate subscriber's particular terminal. The precise format is not defined.
- Presentation: No defined use.
- Screening: No defined use.
- SubAddressString: No defined use.

Hence it is possible to indicate the subscriber and/or the terminal identification. This terminal addressing is implementation specific e.g. subscriber identification may not always be sufficient information to get the capabilities of the terminal.

#### **capabilityScope:in TpTerminalCapabilityScope**

Specifies the scope of the capabilities that the application is interested in. The contents are implementation specific. One possibility is to use the CC/PP definitions as in TpTerminalCapabilities.

#### **criteria:in TpTerminalCapabilityChangeCriteria**

Specifies the trigger conditions for the reports e.g. software or hardware update.

#### *Returns*

**TpAssignmentID**

#### *Raises*

**TpCommonExceptions, P\_INFORMATION\_NOT\_AVAILABLE,  
P\_INVALID\_INTERFACE\_TYPE, P\_INVALID\_CRITERIA, P\_INVALID\_TERMINAL\_ID**

### 8.2.2 Method triggeredTerminalCapabilityStop()

Stop reporting for terminal capability changes that were started by triggeredTerminalCapabilityStartReq().

#### *Parameters*

**assignmentID:in TpAssignmentID**

Specifies the assignment ID for the task to be stopped.

#### *Raises*

**TpCommonExceptions, P\_INVALID\_ASSIGNMENT\_ID**

## 8.3 Interface Class IpAppExtendedTerminalCapabilities

Inherits from: IpInterface.

IpAppExtendedTerminalCapabilities interface is used to send triggered terminal capability reports. It is implemented by the client application developer.

<<Interface>> IpAppExtendedTerminalCapabilities
triggeredTerminalCapabilityReport (assignmentID : in TpAssignmentID, terminals : in TpAddressSet, criteria : in TpTerminalCapabilityChangeCriteria, capabilities : in TpTerminalCapabilities) : void triggeredTerminalCapabilityReportErr (assignmentId : in TpAssignmentID, terminals : in TpAddressSet, cause : in TpTerminalCapabilitiesError) : void

### 8.3.1 Method triggeredTerminalCapabilityReport()

This terminal capability report is issued when the capabilities of the terminal have changed in the way specified by the criteria parameter in the previously invoked triggeredTerminalCapabilityStartReq () method.

#### *Parameters*

**assignmentID: in TpAssignmentID**

Specifies the assignment ID of the report.

**terminals: in TpAddressSet**

Specifies the terminal(s) either by subscriber or terminal ID or both as described for the triggeredTerminalCapabilityStartReq () method.

**criteria: in TpTerminalCapabilityChangeCriteria**

Specifies the criteria that caused the report to be sent.

**capabilities: in TpTerminalCapabilities**

Specifies the capabilities of the terminal. The network may override some capabilities that have been indicated by the terminal itself due to network policies or other restrictions or modifications in the supported capabilities.

### 8.3.2 Method triggeredTerminalCapabilityReportErr()

This method indicates that the requested reporting has failed. Note that errors may concern the whole assignment or just some terminals. In the former case no terminals are specified.

#### *Parameters*

**assignmentId: in TpAssignmentID**

Specifies the assignment ID.

**terminals: in TpAddressSet**

Specifies the terminal(s) either by subscriber or terminal ID or both as described for the triggeredTerminalCapabilityStartReq () method.

**cause: in TpTerminalCapabilitiesError**

Specifies the error that led to the failure.

## 9 State Transition Diagrams

There are no State Transition Diagrams for the Terminal Capabilities SCF.

## 10 Service Properties

The following table lists properties relevant for this SCF.

Property	Type	Description
P_TRIGGERED_REPORTING_SUPPORTED	BOOLEAN_SET	Value = TRUE : The triggered reporting of terminal capabilities is supported by the SCF. Value = FALSE : The triggered reporting of terminal capabilities is not supported by the SCF.

## 11 Terminal Capabilities Data Definitions

The constants and types defined in the following clauses are defined in the *org.csapi.termcap* package.

All data types referenced but not defined in this clause are common data definitions which may be found in ES 204 915-2.

### 11.1 terminalIdentity

Identifies the terminal.

Name	Type	Documentation
terminalIdentity	TpString	Identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy.

### 11.2 TpTerminalCapabilities

This data type is a Sequence of Data Elements that describes the terminal capabilities. It is a structured type that consists of:

Sequence Element Name	Sequence Element Type	Documentation
TerminalCapabilities	TpString	Specifies the latest available capabilities of the user's terminal. This information, if available, is returned as CC/PP headers as specified in W3C (see [6] in ES 204 915-1) and adopted in the WAP UAProf specification (see [9] in ES 204 915-1). It contains URLs; terminal attributes and values, in RDF format; or a combination of both.
StatusCode	TpBoolean	Indicates whether or not the TerminalCapabilities are available.



## 11.3 TpTerminalCapabilitiesError

Defines an error that is reported by the Terminal Capabilities SCF.

Name	Value	Description
P_TERMPCAP_ERROR_UNDEFINED	0	Undefined.
P_TERMPCAP_INVALID_TERMINALID	1	The request can not be handled because the terminal id specified is not valid.
P_TERMPCAP_SYSTEM_FAILURE	2	System failure. The request cannot be handled because of a general problem in the terminal capabilities service or the underlying network.
P_TERMPCAP_INFO_UNAVAILABLE	3	The terminal capability information is not available.

## 11.4 TpTerminalCapabilityChangeCriteria

Defines the type of the terminal capability changes to be reported. The values may be combined by a logical 'OR' function.

Name	Value	Description
P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_UNDEFINED	00h	Undefined.
P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_GENERAL	01h	Any change in the terminal capabilities.
P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_HW_UPDATE	02h	The terminal device hardware has been modified or replaced completely.
P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_SW_UPDATE	04h	The software of the terminal has been updated in any way. Also changes in configuration or preferences may be concerned.
P_TERMINAL_CAPABILITY_CHANGE_CRITERIA_INITIAL	08h	The initial device capabilities reported when monitoring has been started by an application.

## 11.5 TpTerminalCapabilityScopeType

Defines a specific type of the terminal capability scope definition.

Name	Value	Description
P_TERMINAL_CAPABILITY_SCOPE_TYPE_UNDEFINED	0	Undefined.
P_TERMINAL_CAPABILITY_SCOPE_TYPE_CCPP	1	Indicates that the terminal capability scope is expressed as CC/PP headers as specified in W3C (see [6] in ES 204 915-1) and adopted in the WAP UAProf specification (see [9] in ES 204 915-1). It contains URLs; terminal attributes and values, in RDF format; or a combination of both.

## 11.6 TpTerminalCapabilityScope

Defines the Sequence of Data Elements that specify the scope of the terminal capabilities.

Sequence Element Name	Sequence Element Type
ScopeType	TpTerminalCapabilityScopeType
Scope	TpString

---

## 12 Exception Classes

The following are the list of exception classes which are used in this interface of the API.

Name	Description
P_INVALID_TERMINAL_ID	The request can not be handled because the terminal id specified is not valid.

Each exception class contains the following structure.

Structure Element Name	Structure Element Type	Structure Element Description
ExtraInformation	TpString	Carries extra information to help identify the source of the exception, e.g. a parameter name.

---

## Annex A (normative): OMG IDL Description of Terminal Capabilities SCF

The OMG IDL representation of this interface specification is contained in a text file (termcap.idl) contained in archive es\_20491507IDL.zip which accompanies the present document.

This archive can be found in es\_20491507v010101m0.zip which accompanies the present document.

---

## Annex B (informative): W3C WSDL Description of Terminal Capabilities SCF

The W3C WSDL representation of this interface specification is contained in zip file `es_20491507WSDL.zip` which accompanies the present document.

This archive can be found in `es_20491507v010101m0.zip` which accompanies the present document.

---

## Annex C (informative): Java™ API Description of the Terminal Capabilities SCF

The Java™ API realisation of this interface specification is produced in accordance with the Java™ Realisation rules defined in ES 204 915-1. These rules aim to deliver for Java™, a developer API, provided as a realisation, supporting a Java™ API that represents the UML specifications. The rules support the production of both J2SE™ and J2EE™ versions of the API from the common UML specifications.

The J2SE™ representation of this interface specification is provided as Java™ Code, contained in archive 20491507J2SE.zip that accompanies the present document.

The J2EE™ representation of this interface specification is provided as Java™ Code, contained in archive 20491507J2EE.zip that accompanies the present document.

Both these archives can be found in es\_20491507v010101m0.zip which accompanies the present document.

---

## Annex D (informative): Contents of 3GPP OSA R7 Terminal Capabilities

All of the present document is relevant for TS 129 198-7 V7 (Release 7).

---

## Annex E (informative): Description of Terminal Capabilities SCF for 3GPP2 cdma2000 networks

This annex is intended to define the OSA API Stage 3 interface definitions and it provides the complete OSA specifications. It is an extension of OSA API specifications capabilities to enable operation in cdma2000 systems environment. They are in alignment with 3GPP2 Stage 1 requirements and Stage 2 architecture defined in [52], [53] and [54] of ES 204 915-1, clause 2. These requirements are expressed as additions to and/or exclusions from the 3GPP Release 7 specification. The information given here is to be used by developers in 3GPP2 cdma2000 network architecture to interpret the 3GPP OSA specifications.

---

### E.1 General Exceptions

The terms 3GPP and UMTS are not applicable for the cdma2000 family of standards. Nevertheless these terms are used (TR 121 905) mostly in the broader sense of "3G Wireless System". If not stated otherwise there are no additions or exclusions required.

CAMEL and CAP mappings are not applicable for cdma2000 systems.

---

### E.2 Specific Exceptions

#### E.2.1 Clause 1: Scope

There are no additions or exclusions.

#### E.2.2 Clause 2: References

Normative references on TS 123 078 and on TS 129 078 are not applicable for cdma2000 systems.

#### E.2.3 Clause 3: Definitions and abbreviations

There are no additions or exclusions.

#### E.2.4 Clause 4: Terminal Capabilities SCF

There are no additions or exclusions.

#### E.2.5 Clause 5: Sequence Diagrams

There are no additions or exclusions.

#### E.2.6 Clause 6: Class Diagrams

There are no additions or exclusions.

#### E.2.7 Clause 7: The Service Interface Specifications

There are no additions or exclusions.

## E.2.8 Clause 8: Terminal Capabilities Interface Classes

There are no additions or exclusions.

## E.2.9 Clause 9: State Transition Diagrams

There are no additions or exclusions.

## E.2.10 Clause 10: Service Properties

There are no additions or exclusions.

## E.2.11 Clause 11: Terminal Capabilities Data Definitions

There are no exclusions. Additions for Data types for cdma2000 systems are for further study and are not part of this release. (E.g.: terminalIdentity identifies the terminal. It may be a logical address known by the WAP Gateway/PushProxy or any other relevant network elements in cdma2000 network, i.e. HSS).

## E.2.12 Clause 12: Exception Classes

There are no additions or exclusions.

## E.2.13 Annex A (normative): OMG IDL Description of Terminal Capabilities SCF

There are no additions or exclusions.

## E.2.14 Annex B (informative): W3C WSDL Description of Terminal Capabilities SCF

There are no additions or exclusions.

## E.2.15 Annex C (informative): Java™ API Description of Terminal Capabilities SCF

There are no additions or exclusions.



---

## Annex F (informative): Record of changes

The following is a list of the changes made to the present document for each release. The list contains the names of all changed, deprecated, added or removed items in the specifications and not the actual changes. Any type of change information that is important to the reader is put in the final clause of this annex.

Changes are specified as changes to the prior major release, but every minor release will have its own part of the table allowing the reader to know when the actual change was made.

---

### F.1 Interfaces

#### F.1.1 New

Identifier	Comments
<b>Interfaces added in ES 204 915-7 version 1.1.1 (Parlay 6.0)</b>	

#### F.1.2 Deprecated

Identifier	Comments
<b>Interfaces deprecated in ES 204 915-7 version 1.1.1 (Parlay 6.0)</b>	

#### F.1.3 Removed

Identifier	Comments
<b>Interfaces removed in ES 204 915-7 version 1.1.1 (Parlay 6.0)</b>	

---

### F.2 Methods

#### F.2.1 New

Identifier	Comments
<b>Methods added in ES 204 915-7 version 1.1.1 (Parlay 6.0)</b>	

#### F.2.2 Deprecated

Identifier	Comments
<b>Methods deprecated in ES 204 915-7 version 1.1.1 (Parlay 6.0)</b>	

### F.2.3 Modified

Identifier	Comments
Methods modified in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

### F.2.4 Removed

Identifier	Comments
Methods removed in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

---

## F.3 Data Definitions

### F.3.1 New

Identifier	Comments
Data Definitions added in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

### F.3.2 Modified

Identifier	Comments
Data Definitions modified in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

### F.3.3 Removed

Identifier	Comments
Data Definitions removed in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

---

## F.4 Service Properties

### F.4.1 New

Identifier	Comments
Service Properties added in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

## F.4.2 Deprecated

Identifier	Comments
Service Properties deprecated in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

## F.4.3 Modified

Identifier	Comments
Service Properties modified in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

## F.4.4 Removed

Identifier	Comments
Service Properties removed in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

---

## F.5 Exceptions

### F.5.1 New

Identifier	Comments
Exceptions added in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

### F.5.2 Modified

Identifier	Comments
Exceptions modified in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

### F.5.3 Removed

Identifier	Comments
Exceptions removed in ES 204 915-7 version 1.1.1 (Parlay 6.0)	

---

## F.6 Others

None.

---

## History

<b>Document history</b>		
V1.1.1	February 2008	Membership Approval Procedure MV 20080425: 2008-02-26 to 2008-04-25