

WAP WTA

Draft Version 30-Apr-1998

Wireless Application Protocol Wireless Telephony Application Specification

Disclaimer:

The document is subject to change without any notice.

Contents

1	SCOPE	4
2	DOCUMENTS STATUS	4
2.1	COPYRIGHT NOTICE.....	4
2.2	ERRATA.....	4
2.3	COMMENTS.....	4
3	REFERENCES	4
3.1	NORMATIVE REFERENCES.....	4
4	DEFINITIONS AND ABBREVIATIONS	5
4.1	DEFINITIONS.....	5
4.2	ABBREVIATIONS.....	7
5	WTA OVERVIEW	8
5.1	SERVER CENTRIC AND CLIENT CENTRIC MODE.....	8
5.2	USER-AGENT CHARACTERISTICS.....	9
5.3	WTAI FUNCTION CALLS.....	9
5.4	WTA EVENTS.....	9
6	SUPPORTED CONTENT FORMATS	10
6.1	PERFORMANCE REQUIREMENTS.....	11
6.2	CONTENT SERIALISATION.....	11
6.3	SERVER CENTRIC MODE.....	11
6.4	CLIENT CENTRIC MODE.....	12
7	WTA URI'S AND URL'S	13
7.1	USE OF URL'S.....	13
8	THE USER-AGENT STATE MODEL	14
8.1	THE USER-AGENT CONTEXT.....	14
8.2	TERMINATION OF CONTEXT.....	14
8.3	VARIABLES.....	14
9	WTA EVENTS AND NAVIGATION	15
9.1	PRECEDENCE OF EVENT HANDLERS.....	15
9.2	WTA EVENT TABLE.....	16
9.2.1	<i>Programming the Event Table</i>	16
9.2.2	<i>WTA Event Table Binding</i>	17
9.3	WTA EVENT BINDINGS IN USER-AGENT CONTEXT.....	18
10	USER-AGENT SEMANTICS	19
10.1	LOW-MEMORY BEHAVIOUR.....	19
10.1.1	<i>Limited History</i>	19
10.1.2	<i>Limited User-agent Context Size</i>	19
10.2	ERROR HANDLING.....	19
10.3	UNKNOWN DTD.....	19
11	WTA SESSION MANAGEMENT	20
11.1	START-UP OF A WTA SESSION.....	20

- 11.2 TERMINATION OF A WTA SESSION 22
- 11.3 RELIABLE CONTENT PUSH 22
- 12 WTA REFERENCE INFORMATION..... 23**
- 12.1 DOCUMENT IDENTIFIERS..... 23
 - 12.1.1 SGML Public Identifier..... 23
 - 12.1.2 WTA Event Media Type..... 23
- 12.2 DOCUMENT TYPE DEFINITION (DTD)..... 23
- 13 WTA EVENT BINARY ENCODING..... 24**
- 13.1 EXTENSION TOKENS 24
 - 13.1.1 Global Extension Tokens..... 24
 - 13.1.2 Tag Tokens 24
 - 13.1.3 Attribute Tokens 24
- 13.2 ENCODING SEMANTICS 25
 - 13.2.1 Encoding Variables..... 25
 - 13.2.2 Document Validation 25
- 13.3 NUMERIC CONSTANTS 25
 - 13.3.1 Tag Tokens 25
 - 13.3.2 Attribute Start Tokens 25
 - 13.3.3 Attribute Value Tokens 25
- 13.4 WTA ENCODING EXAMPLES 26
 - 13.4.1 WTA Event 26
 - 13.4.2 WTA Event Table 27
- 14 WTA EXAMPLES..... 28**
- 14.1 PROVISIONING 28
- 14.2 INCOMING CALL 29
- 14.3 VOICE MAIL..... 30

1 Scope

Wireless Application Protocol (WAP) is a result of continuous work to define an industry wide specification for developing applications over wireless communication networks. The scope for the WAP working group is to define a set of standards to be used by service applications. The wireless market is growing very quickly and reaching new customers and services. To enable operators and manufacturers to meet the challenges in advanced services, differentiation and fast/flexible service creation, WAP defines a set of protocols in transport, session and application layers. For additional information on the WAP architecture, refer to "*Wireless Application Protocol Architecture Specification*" [WAP].

This specification defines the Wireless Telephony Application (WTA). The WTA user-agent is an extension of the WAE user-agent using WTA Interface [WTAI]. WTA is intended for use in specifying Wireless Telephony Applications that interface with local and network telephony infrastructure.

WTA interfaces can be broken down to three different types. The first two types, Network Common and Network Specific, are reserved for the network operators. This is because the mobile network operators control and maintain the services for users on their mobile network. The third type Public, is a limited set of WTA functions, such as initiating a mobile phone call, available to content from any content developer [WTAI].

Wireless Telephony Applications have four main goals:

1. Advanced end user services, WTA makes it possible for network operators to provide advanced services with a consistent interface towards the end users.
2. Increased utilisation of network, network operators can utilise WTA to increase the use of the network services.
3. Interoperability, WAP Applications written using the WTA interface can execute on a variety of telephony devices. Interoperability across Terminals!
4. Network-independent applications, WTA content developers write telephony applications that span various networks that use different protocols.

2 Documents Status

This document is available online in the following formats:

- PDF format at URL, <http://www.wapforum.org/>.

2.1 Copyright Notice

© Copyright Wireless Application Forum Ltd, 1998 all rights reserved.

2.2 Errata

Known problems associated with this document are published at <http://www.wapforum.org/>.

2.3 Comments

Comments regarding this document can be submitted WAP in the manner published at <http://www.wapforum.org/>.

3 References

3.1 Normative references

- [RFC1738] "Uniform Resource Locators (URL)", T. Berners-Lee, et al., December 1994. URL: <ftp://ds.internic.net/rfc/rfc1738.txt>
- [RFC1630] "Uniform Resource identifiers (URI)", T. Berners-Lee, et al., June 1994. URL: <ftp://ds.internic.net/rfc/rfc1630.txt>
- [WAE] "Wireless Application Environment Specification", WAP Forum, 1998. URL: <http://www.wapforum.org/>
- [WAEARCH] "Wireless Application Environment Architecture Overview", WAP Forum, 1998. URL: <http://www.wapforum.org/>
- [WAP] "Wireless Application Protocol Architecture Specification, version 0.9", WAP Forum, 1997. URL: <http://www.wapforum.org/>
- [WML] "Wireless Markup Language", WAP Forum, 1997. URL: <http://www.wapforum.org/>
- [WMLScript] "WMLScript Language Specification", WAP Forum, 1998. URL: <http://www.wapforum.org/>
- [WAESTdLib] "WMLScript Standard Libraries Specification", WAP Forum, 1997. URL: <http://www.wapforum.org/>
- [WSP] "Wireless Session Protocol Specification", WAP Forum, 1997. URL: <http://www.wapforum.org/>
- [WTAI] "Wireless Telephony Application Interface Specification", WAP Forum, 1997. URL: <http://www.wapforum.org/>
- [XML] "Extensible Markup Language (XML), W3C Recommendation 10-February-1998, REC-xml-19980210", T. Bray, et al, February 10, 1998. URL: <http://www.w3.org/TR/REC-xml>

4 Definitions and abbreviations

All non-trivial abbreviations and definitions used in this document are listed in the following sections. The definitions section includes description of general concepts and issues that may be fully defined in other documents. The purpose of this section is merely to advise the reader on the terminology used in the document.

4.1 Definitions

The following are terms and conventions used throughout this specification.

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described by [RFC2119].

Author - an author is a person or program that writes or generates WML, WMLScript or other content.

Card - a single WML unit of navigation and user interface. May contain information to present on the screen, instructions for gathering user input, etc.

Clear – used in conjunction with WTA context means that all variables stored in the WTA user context is removed.

Client - a device (or application) that initiates a request for connection with a server.

Client Centric – external events, e.g. network events, are processed by the client portion of the WTA application..

Content - subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent in response to a user request.

Content Encoding - when used as a verb, content encoding indicates the act of converting a data object from one format to another. Typically the resulting format requires less physical space than the original, is easier to process or store, and/or is encrypted. When used as a noun, content encoding specifies a particular format or encoding standard or process.

Content Format – actual representation of content.

Content Generator - a service that generates or formats content. Typically, content generators are on origin servers.

Deck - a collection of WML cards. A WML deck is also an XML document. May contain WMLScript.

Device - a network entity that is capable of sending and receiving packets of information and has a unique device address. A device can act as either a client or a server within a given context or across multiple contexts. For example, a device can service a number of clients (as a server) while being a client to another server.

Device OS – the standard operating system in a device. Not part of what is specified by WAP.

Implicit Variables – Variables that are created by the user-agent itself because of an incoming WTA event.

Origin Server - the server on which a given resource resides or is to be created. Often referred to as a web server or an HTTP server.

Resource - network data object or service that can be identified by a URL. Resources may be available in multiple representations (e.g. multiple languages, data formats, size, and resolutions) or vary in other ways.

Reset – used in conjunction with WTA context meaning that the state of the WTA user-agent is set to a state with an empty history stack. Implementation specific state is set to a well-known state.

Server - a device (or application) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client.

Server Centric – content from different sources is terminated in the server.

Terminal - a device typically used by a user to request and receiving information. Also called a mobile terminal or mobile station.

User - a user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content.

User Agent - a user agent (or content interpreter) is any software or device that interprets resources. This may include textual browsers, voice browsers, search engines, etc.

Web Server - a network host that acts as an HTTP server.

WML - the Wireless Markup Language is a hypertext markup language used to represent information for delivery to a narrowband device, e.g. a mobile phone.

WMLScript - a scripting language used to program the mobile device. WMLScript is an extended subset of the JavaScript™ scripting language.

WTA Context - the complete set of variables, with content and the state of the WTA user-agent.

WTA Event Context Binding – a WTA event name specified using the type attribute in the WML "ONEVENT" construct [WML].

WTA Event – A notification, in the form of content, that conveys a change of state of the sender. A WTA event can be received from a WTA Originating Server or be generated from events occurring in the mobile network, i.e. Incoming call.

WTA Event Table – A specific WTA content format with an ordered sequence of WTA event table bindings. The WTA event table is stored persistently in a device using a device dependent format.

WTA Event Table Binding – a WTA event name with an associated URL and an optional list of parameter names.

WTA Event Handler – the entity that is used for the event matching. Can be the WTA context, the WTA event table or the device OS.

WTA event matching – the process where a WTA event name is compared with a WTA event name stored as a WTA event table binding or in the WTA context using WTA event context bindings. If the WTA event name is equal to the stored WTA event name then there is a match.

XML - the Extensible Markup Language is a World Wide Web Consortium (W3C) proposed standard for Internet markup languages, of which WML is one such language. XML is a restricted subset of SGML.

4.2 Abbreviations

For the purposes of this specification, the following abbreviations apply.

API	Application Programming Interface
BNF	Backus-Naur Form
CGI	Common Gateway Interface
ECMA	European Computer Manufacturers Association
ETSI	European Telecommunication Standardisation Institute
GSM	Global System for Mobile Communication
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IANA	Internet Assigned Number Authority
MSC	Mobile Switching Centre
MSISDN	Mobile Station International Subscriber Device Number
RFC	Request For Comments
SGML	Standardised Generalised Markup Language
URI	Uniform Resource Identifier
URL	Uniform Resource Locator [RFC1738]
WAE	Wireless Application Environment
WAP	Wireless Application Protocol [WAP]
WSP	Wireless Session Protocol [WSP]
WTA	Wireless Telephony Applications
WTAI	Wireless Telephony Applications Interface
WTP	Wireless Transaction Protocol
WWW	World Wide Web
W3C	World Wide Web Consortium
XML	Extensible Markup Language [XML]

5 WTA Overview

WTA enables content written in WML and WMLScript to utilise telephony features in the device and the mobile network. The WTA server, the mobile network and the WTA client can be thought of as a single application with parts in the client and the WTA server. The WTA server acts as the principal content generator. The WTA server may be connected to the mobile network where it could have the means to control the mobile network services. Content may be customised by the WTA server and downloaded to the client.

WTA extends the basic WAE application model in three ways:

1. WTA provides a means for a WTA server to push content to the device.
2. WTA provides a means for mobile network events to trigger the rendering of content in the device.
3. WTA provides telephony functions on the device that can be accessed from WML or WMLScript.

The WSP push feature [WSP] is used by the WTA server to push down content to the WTA client. While content can be essentially anything, there is two fundamentally different types of content formats: standard WAE content formats, such as WML, WMLScript or WBMP, and the specific WTA content format called WTA event. The WTA framework allows a flexible implementation for the client that allows it to support pushed standard content, WTA event or both. The WTA server must decide what features to use based on the user-agent characteristics and the profile specified for the specific WTA User-agent [WAE].

5.1 Server centric and Client centric mode

The fundamental differences, between pushing a WTA event or content can best be illustrated using the pictures below. The co-ordinating network element has been used for naming the particular model. The two important network elements in this case are the WTA server and the WTA client. In both models, the WTA server acts as the content generator towards the WTA client.

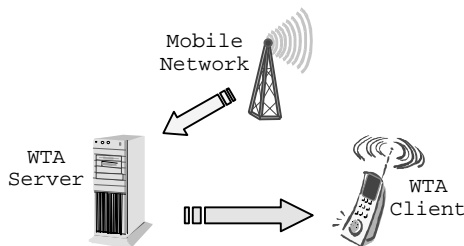


Figure 1, server-centric mode

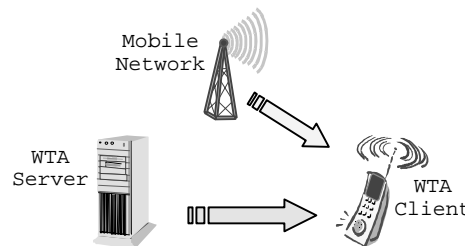


Figure 2, client-centric mode

1. **Server-centric mode (Figure 1):**
The WTA server handles events, such as incoming call, that occurs in the mobile network and provides the WTA client with content that is used for handling the specific task. The WTA server deploys content customised for the occurring event.
2. **Client-centric mode (Figure 2):**
The WTA client handles events, such as incoming call, received from the mobile network. Event bindings in the WTA client can be used to associate the event to executable content that is used for handling the specific task.

By combining the server and client centric model in the same architecture, the operator can take advantage of the features that best fits the particular type of mobile network and type of applications.

5.2 User-Agent Characteristics

The WAP proxy is informed of the available WTA features such as available WTAI Libraries [WTAI] and the selected WTA network-programming model, i.e. server or client centric mode

5.3 WTAI Function Calls

Provides access to features in the device and the mobile network [WTAI]. An example of local device features is logical indications for call control . Mobile network features include call control and network text.

5.4 WTA Events

The picture below illustrates WTA event transition. WTA event transition is the process used by the WTA user-agent for handling the transformation of network event to a WTA event and then to an appropriate action by the client (i.e. ringing on the event of an incoming call. WTA events are then matched to the appropriate WTA event handler depending on the state of the WTA user-agent and the current WTA context if any.

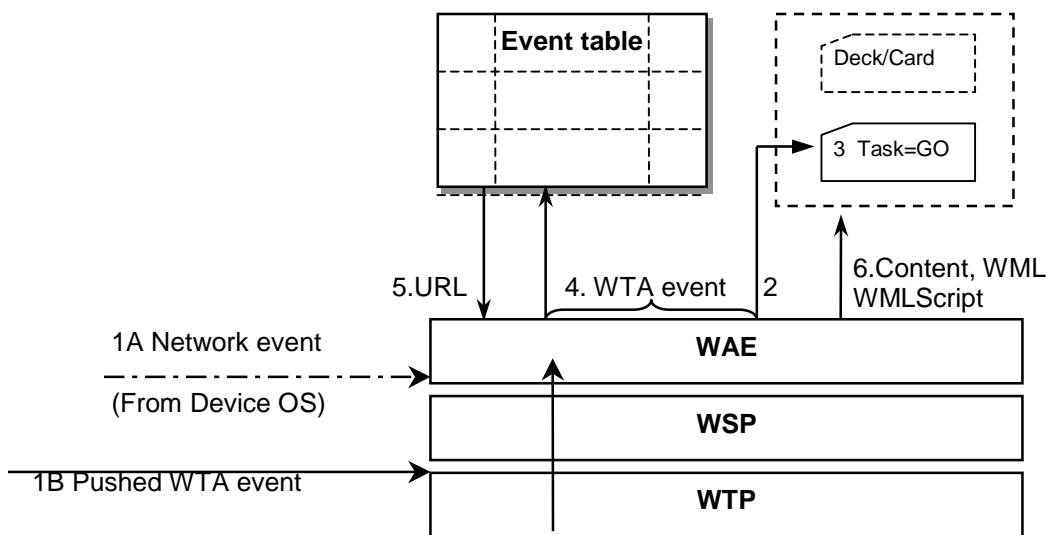


Figure 3, WTA event transformation

The two ways in which a client can receive a WTA event are illustrated in figure 3. The first way is via the device operating system, the OS detects an event and then passes that event to the WAE basic services, this is denoted with an A as a suffix. The second way, denoted with a B as a suffix, is through a pushed WTA event which is received from the WTA server. Network events, as specified by [WTAI], are converted into the abstract WTA event form.

Table 1, Event flow walkthrough

- 1A The Device OS receives a network event and converts it to a WTA event which is injected into WAE.
- 1B The WTA event content format is received from the WTA server using the WSP push service [WSP].
- 2 In case of a user-agent context the WTA event must first be matched to any existing WML event bindings.
- 3 In case of a WML event bindings with task=go the URL locates the resource that must be loaded [WML].
- 4 No match with the WML event binding means the WTA event. must be tested for a match with the event table.
- 5 A match must return the associated URL. The URL is extracted from the WTA event table binding.
- 6 The WTA user-agent is restarted with a new context and then loads the content located by the URL¹. Any values received, with the WTA event, must be stored in the variables identified by the parameters list.

¹ A URL, stored in the event table, can be relative i.e. a card name, in which case the base URL, must be provided with the event table.

6 Supported Content Formats

This section describes the specific types of content formats that must be supported and the required behaviour of the WTA client depending on the mode, client centric or server centric when receiving the content.

The following three fundamental methods for content transfer are described:

- Mobile Network Event:**
 The representation of the network specific signalling that can be handled [WTAI]. The WTA user-agent encounters the network event in an abstract form of the WTA event. Abstract as no physical WTA event content is needed.
- Acknowledged Content:**
 Content that must be acknowledged to the sending party, example is the event table.
- Unacknowledged Content:**
 Content that does not require an acknowledge, example is a WML deck

The following diagram shows the content formats that must be supported by the WTA client depending on the WTA client mode, server centric or client centric. The content formats supported in the server centric mode must also be supported by the client centric mode.

The diagram is constructed based on the domains of content formats for the client centric and server centric mode. The server centric content formats are thus also contained in the client centric domain. The opposite is not true, i.e. there is no support for event table or WTA event in the server centric domain.

Methods for content transfer are rendered in a bold font. Arrows point to the content formats that could be transferred using the specified method. *Italic font* denotes the processing that will take place for content formats that originates from events (WTA events or mobile network events). This is only a logical diagram as network events, in reality must be converted to an abstraction of a WTA event prior to be processed by the user-agent or the event table.

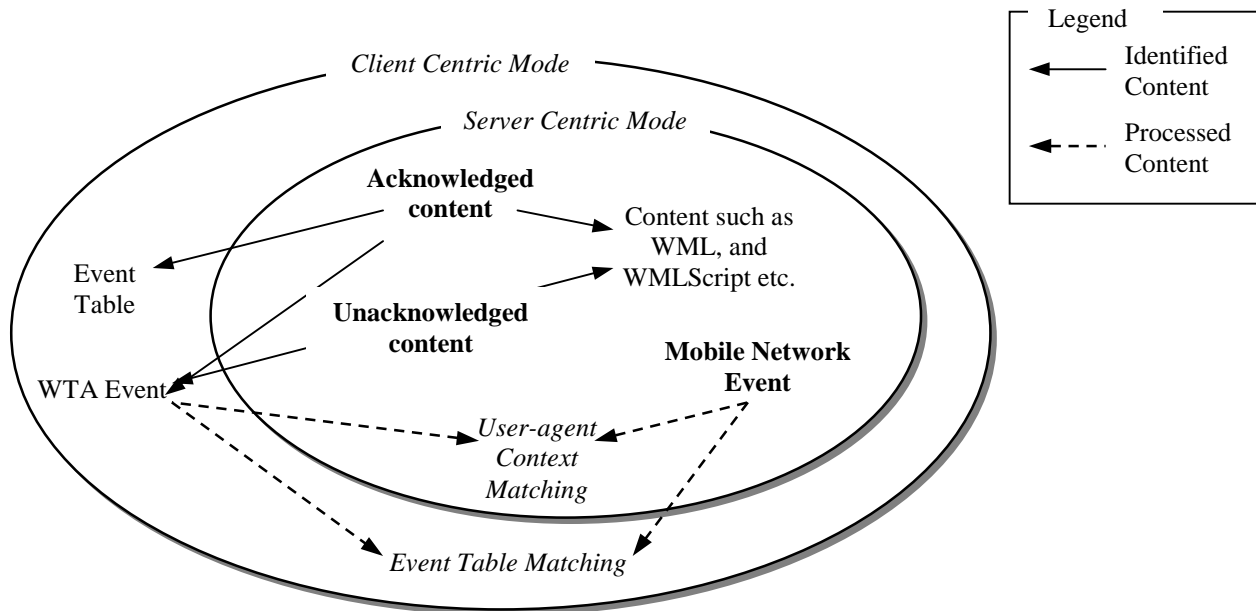


Figure 4, WTA content formats relational diagram

6.1 Performance requirements

The terms “immediately” and “unattended” used below are used to describe the expected response for a particular implementation in the WTA client and the WTA server. The term “**Immediately**” means that the user-agent must cancel the execution of any other content and start the processing of the new content. The term “**unattended**” means that from the WTA servers perspective there has been no “acknowledge” response from the WTA client during a predefined timeout period. The timeout period is dependent on the bearer used. Refer to the WAP architecture document [WAPARCH], for recommendations on bearer characteristics such as the timeout for acknowledged content to the WTA client.

6.2 Content Serialisation

The model for handling content must apply an order of processing that ensures all content to be handled in the order of reception. No other content must be handled until the processing of the current content is concluded. A WTA event is regarded concluded, as soon as the event have been resolved into a URL or by other means, such as the WML event bindings, where it is bound to a specific task, have begin execution. Handling of content, such as WML and WMLScript are regarded concluded as soon as the content has been loaded into the context and the execution has started.

6.3 Server Centric Mode

In this mode WTA event and event table must no be sent to the WTA client from the WTA server. The following rules state the required behaviour for a WTA user-agent and WTA server in server centric mode:

Required steps for handling a Mobile network event:

1. The network event must be serviced immediately after reception and conversion to WTA event.
2. If a user-agent context exist, “Event matching” must be applied in order to find a matching WML event.binding stored in the user-agent context, i.e. a WML element “ONEVENT” with a matching event name.

Required steps for handling acknowledged content:

1. The content must be serviced immediately by the WTA user-agent.
2. When the content has been identified by the WTA user-agent, the result code indicating “success” must be acknowledged [WSP] to the WTA server.
3. For content that can not be identified by the WTA user-agent, the result code indicating “failure” must be acknowledged [WSP] to the WTA server.
4. For unattended content, the WTA server must apply the same rule as if the content was acknowledged with the result code indicating “failure”.

Required steps for handling unacknowledged content:

1. The content must be indicated to the end-user. There must be a possibility for the end-user to detect content that eventually needs to be attended. How an end-user indication might look like is up to the implementation and is not specified by WAP.
2. The content must not be processed unless the context in the WTA user-agent has been terminated. In other words, the new content must not be allowed to interrupt any other content that is executed by the user-agent. The end-user may decide to terminate any current content, and context, in the user-agent and instead attend the new content.

6.4 Client Centric Mode

The same rules applies for a WTA user-agent in client centric mode as for the server centric mode with regards to handling mobile network event and content such as WML and WMLScript. In addition, the client centric mode also requires that the WTA user-agent must support WTA event and the event table.

The following rules state the required behaviour for a WTA client in client centric mode:

Required steps for handling event table:

1. The event table must be serviced immediately by the WTA user-agent:
2. When the event table has been successfully set-up by the WTA user-agent, the result code indicating “success” must be acknowledged [WSP] to the WTA server. A successful set-up means that the event table was possible to verify with regards to the required elements in the encoded form. Refer to section 13. The new event table is effective at the same time an acknowledgement is sent back.
3. For an event table that can not be set-up by the WTA user-agent, due to reception of a mobile network event or a failure in the verification, the result code indicating “failure” must be acknowledged [WSP] to the WTA server.
4. For unattended event table content, the WTA server must apply the same internal rules as if the event table content was acknowledged with the result code indicating “failure”. The internal rules is specific to any implementation of the WTA server.

Required steps for handling acknowledged WTA event:

1. The WTA event must be serviced immediately by the WTA user-agent
2. Event matching determines if a match can be found in an existing WTA user-agent context. If a WTA event is verified to be correct with regards to the required elements in the encoded form. Refer to section 13 and there is a match to the event bindings, the result code indicating “success” must be acknowledged [WSP] to the WTA server.
3. If not match can be found for a existing WTA user-agent context or the WTA event was not possible to verify with regards to the required elements in the encoded form. Refer to section 13., event matching must be applied to the event table. If a match can be found the result code indicating “success” must be acknowledged [WSP] to the WTA server.
4. If a match can not be found in the event table, the result code indicating “failure” must be acknowledged [WSP] to the WTA server.
5. For a WTA event that can not be handled by the WTA user-agent, due to reception of a mobile network event or a failure in the verification, the result code indicating “failure” must be acknowledged [WSP] to the WTA server.
6. For unattended WTA event, the WTA server must apply the same rule as if the WTA event was acknowledged with the result code indicating “failure”.

Required steps for handling unacknowledged WTA event:

1. The WTA event must be serviced immediately by the WTA user-agent
2. The WTA event is handled according to the procedures described in section 9

7 WTA URI's and URL's

Wireless devices come in many flavours with regards to available bearers and types of telephony features. There are many standards, such as GSM, that supports device access to telephony features the mobile network. URI's forms a unifying naming model for how to identify features independently of the internal structure in the device and the mobile network.

In WTA URI's are used in the following situations:

- When specifying the identity of local resources like logical indicators [WTAI].
- When specifying telephony features like set-up of mobile originated call [WTAI].

7.1 Use of URL's

The WTA user-agent is based on the WAE standard user-agent and as such the same rule applies for specifying and using URL's [WML].

In WTA the URL's can also occur in the following situations:

- When specifying URL's in the WTA event table.
- When specifying well-known resources in the WTA Server like the event table.

8 The User-agent State Model

The WTA user-agent includes the same basic support for managing user-agent state as the standard WAE user-agent [WML].

The WTA user-agent also includes support for managing user-agent state including:

- Associated state of a specific call with the user-agent context. WTAI functions such as accept or set-up calls, must be associated with a behaviour how the call is handled in case the WTA user-agent context is intentionally terminated [WTAI]
- Implementation-dependent state - other state relating to the particulars of the user agent implementation and it's behaviour. Examples are the current power save status, memory used and communication settings for SMS etc.

8.1 The User-agent Context

WTA User-agent is based on the WAE standard user-agent and as such the same principles apply for handling a user-agent context. In addition, the WTA user-agent also includes support for termination of the browser context and implicit creation of variables due to reception of WTA event parameters.

8.2 Termination of Context

The WTAI function "exit" [WTAI] indicates that the user-agent context must be "terminated" and the following operations must be performed by the WTA user-agent:

- Active calls are handled according to the mode of the call, i.e. the association with the life span of the context [WTAI].
- Any processing of content such as WML, WMLScript is terminated.
- The browser context is removed and all data stored in the context is consequently discarded.

8.3 Variables

The WTA user-agent is based on the WAE standard user-agent [WAE] and as such the same rules apply for variables as used with the standard WAE user-agent [WML].

The WTA user-agent also includes support for implicit creation of variables in the following situations:

1. Optional parameters received with a WTA event bound to the user-agent context with a WTA event binding can be parameterised, see section 9.3, and are accessible to the user-agent context.
2. Optional parameters received with a WTA Event that is matched against the Event table in the form of a WTA event table binding can be parameterised, see section 9.2, and are accessible to the user-agent context.

9 WTA Events and Navigation

The WTA event contains data to be stored in the current WTA context. The procedure for doing this is referred to using the term “WTA event binding”. How the “WTA event binding” is performed depends on the state of the WTA user-agent.

There are three distinct states when a WTA event can be handled:

1. **The user-agent context matches a WTA event:**
The user-agent context contains an event binding matching with the WTA event.
2. **No active user-agent context or WTA event match:**
The WTA event table contains a WTA event binding that matches the WTA event.
3. **No active user-agent context or WTA event match with the event table bindings:**
If the WTA event originates from a mobile network event the WTA event may be forwarded to the device OS.

9.1 Precedence of event handlers

The following steps must be carried out in the specified order.

Table 2, Precedence of event handlers

1. **User-agent context:**
If there is a user-agent context then event matching must be used to determine if the WTA event matches with an event binding in the user-agent context.
2. **Event Table:**
If there is a event table then event matching must be used to determine if the WTA event matches with an event binding in the event table.
3. **Device OS:**
WTA events, originating from the mobile network, i.e. converted from a network event into a WTA event, must be forwarded to the device OS. WAP does not however require the device OS to handled events.

9.2 WTA Event Table

9.2.1 Programming the Event Table

The Client can be programmed, using the Event Table, to act on incoming events from the mobile network or the WTA Server. WTA events that are received are transformed into URL's that can be executed by the WTA user-agent. The WTA event table is programmed using the WTA event content format that must be pushed from the WTA Server and requested. During start-up of a WTA session in client centric mode.

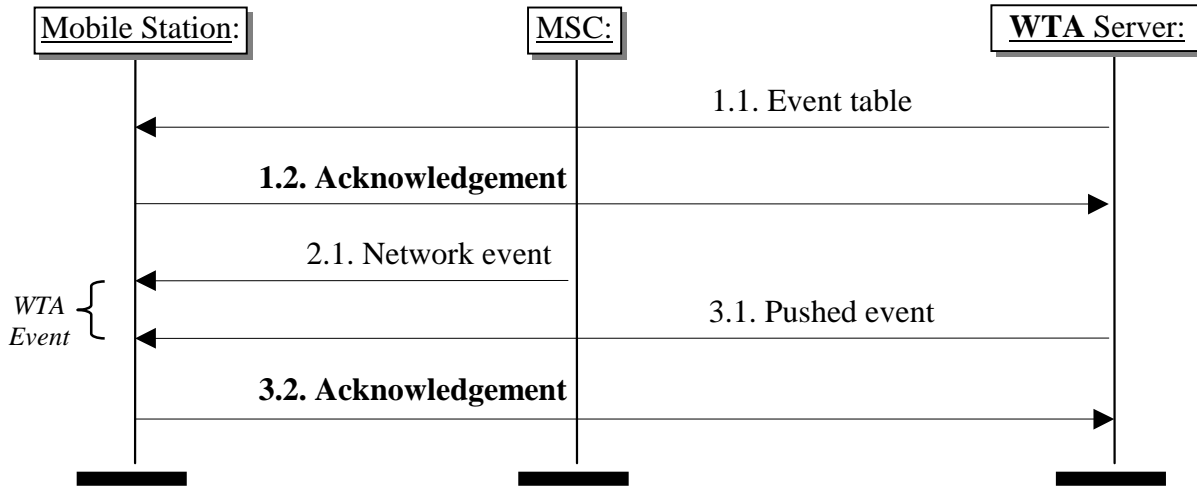


Figure 5, Event table programming

Example 1. :

Server pushes an event table to be set-up by the WTA client

Example 2. :

Network event/message occurs. In this case, the WTA client converts the actual network event to the more general WTA event. If the event binding exists in the event table, any optional parameters are stored in a new user-agent context and the associated URL is called.

Example 3.:

Server pushes a WTA event. The WTA event is handled in the same way as the converted network event.

9.2.2 WTA Event Table Binding

The WTA event table is consulted when there is no current WTA user-agent context or there was no match in the WTA user-agent context for that particular WTA event. The following steps must be carried out.

1. Check the event table for a matching event binding with the WTA event.
2. In case of match the user agent calls the indicated URL stored in the event table. The WTA content is loaded into a clean context. I.e. no previous variables or state is associated with the user-agent context.
3. The WTA event table contains the names of the variables that must be assigned to the values of the parameters received with the WTA event.

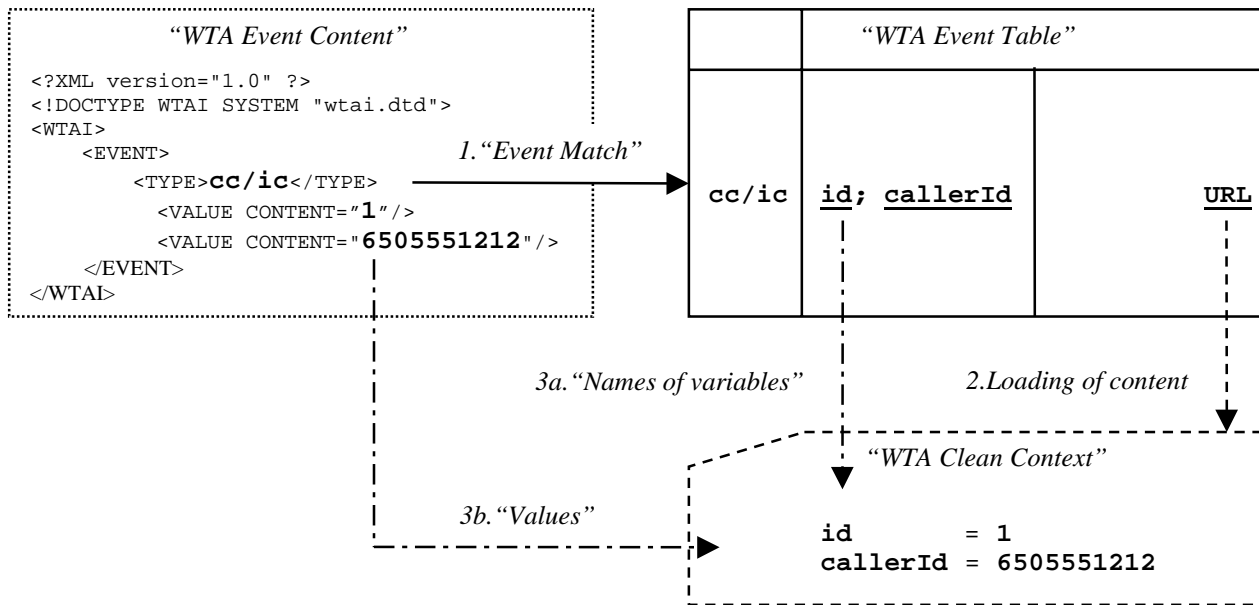


Figure 6, Binding Event data to new user-agent context

9.3 WTA Event Bindings in user-agent Context

Data received with the WTA event can be stored in the user-agent context. The WML construct for handling event defines the syntax how to specify the WTA event bindings using the WML markup. See details in [WML].

The following rules must be applied when using event bindings with the user-agent context:

1. The event bindings stored in the WTA user-agent context must be consulted for a match with the particular WTA event.
2. Optional parameters received with the WTA event must be made available to the user-agent context in the form of implicitly created variable names.
3. The WTA user-agent context must assign the implicit variables to other variables or use the implicit variables as parameters when calling a URL.

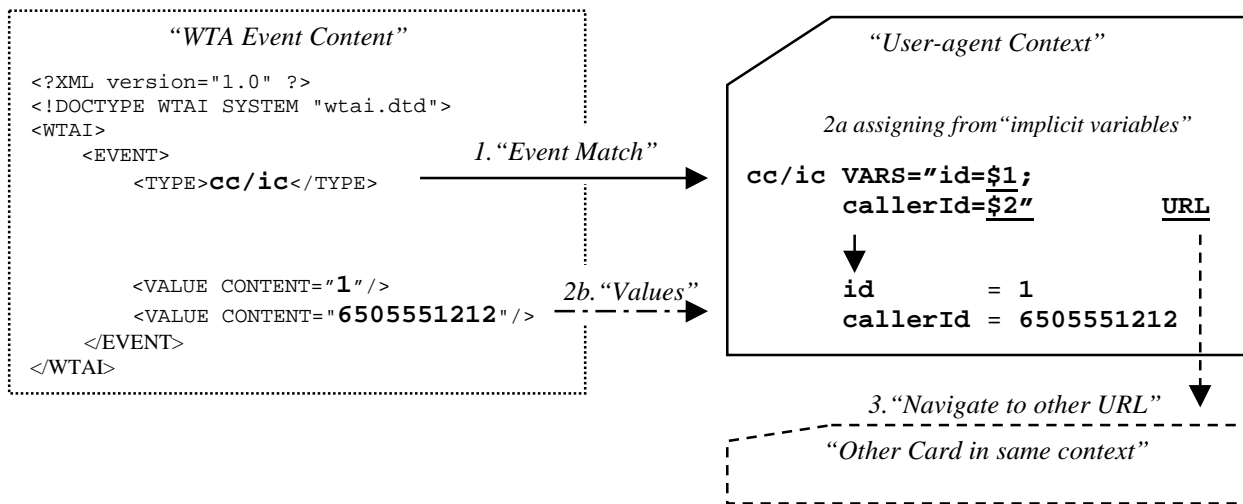


Figure 7, Binding Event data to user-agent context

10 User-Agent Semantics

The WTA user-agent is based on the standard WML User-agent, using WML and WMLScript. WAE identifies content for the WTA user-agent based on the fact that it was received through a dedicated WTP port in the device for WTA access. Content, executed by the WTA user-agent is fully qualified for access to telephony features in the device and the mobile network using the WTAI telephony functions Interface [WTAI].

WTA network events are the means for the WTA user-agent to receive the response from the mobile network. WTA events or other content can also be received from the WTA server using the WTA content push feature. WTA network event is derived from the network-to-client signalling and is recast into the more generic and abstract WTA event.

A set of user-agent characteristics defines the required behaviour for the WTA user-agent:

1. There must be only one WTA user-agent context simultaneously active at a specific time.
2. Reception of “pushed” content [WSP] must be supported by the WTA user-agent in either the server centric or client centric mode. WTA event only with the client centric mode.
3. Any existing WTA user-agent context must be removed prior to processing any pushed content except for WTA event.
4. The WTA user-agent context must be terminated when instructed by the content.
5. There is a predefined order of precedence that must be followed when handling WTA events.

10.1 Low-Memory Behaviour

WTA is targeted at devices with limited hardware resources, including significant restrictions on memory size. It is important that the author have a clear expectation of device behaviour in error situations, including those caused by lack of memory.

10.1.1 Limited History

The user agent may limit the size of the history stack (i.e. the depth of the historical navigation information). In the case of history size exhaustion, the user agent should delete the least-recently-used history information.

It is recommended that all user agents implement a minimum history stack size of ten entries.

10.1.2 Limited User-agent Context Size

In some situations, it is possible that the author has defined an excessive number of variables in the browser context, leading to memory exhaustion.

In this situation, the user agent should attempt to acquire additional memory by reclaiming cache and history memory as described in sections 10.1.1. If this fails and the user agent has exhausted all memory, the user should be notified of the error.

10.2 Error Handling

Conforming user agents must enforce error conditions defined in this specification.

10.3 Unknown DTD

A WTA event content format encoded with an alternate DTD may include elements or attributes that are not recognised by certain user agents. In this situation, a user agent should handle the WTA event content format as if the unrecognised tags and attributes were not present.

11 WTA Session Management

This section describes the required behaviour for set-up and termination of a WTA session. A WTA session is used by the WTA user-agent to facilitate interaction with the WTA server and the mobile network. The term WTA session is used in the following section to denote a WSP session [WSP] over a secure port [WTLS] connecting the WTA client with the WTA server [WAEARCH].

Setting up a WTA session also includes the set-up procedure for the event table when the WTA client supports the client centric mode see section [xx.yy]. The WTA server must then use a reliable connection [WSP] for download of the event table. For additional information see the WSP [WSP] and the WTLS [WTLS] specifications.

The following common WSP and WTLS services are used by the WTA client to start-up a WTA session:

- Start-up a new WSP session
- Request content (event table)
- Start-up a WTLS, secure session
- Termination of a WSP session

11.1 Start-up of a WTA session

The start-up makes it possible for a WTA server to gracefully detect that a WTA client can communicate with the WTA server and the mobile network.

The following rules apply for the WTA server when determining that the WTA session has been started:

1. An indication from the WAP gateway that a WTA session to the WTA client has been established.
2. A "successful" content exchange, pushed or pulled [WSP], i.e. an old session was resumed.

For the start-up of the WTA session the following information must be provided by the implementation:

1. Required bearer (based on preference or availability)
2. Required level of security, bearer dependent [WAPARCH]
3. The WTA server address (bearer specific, for example MSISDN)
4. The well known WTA server port number [WSP]
5. The WTA user-agent characteristics [WAE]

Starting up a WTA user-agent must include the following steps:

(The keyword "conditional" used below denotes a step used when supporting the client centric mode)

1. Request a new WTA session:

A request for a secure WSP session [WSP] to be used with WTA, is sent to the WSP session layer.

2. Check for existing Event table (conditional):

The WTA client must check for the presence of an existing WTA event table

Missing event table (i.e. due to memory loss or intentionally removed from device):

Send a request for a new event table to the WTA server. The request must omit the http header "time of last modification" or include an empty "time of last modification" header.

Existing event table:

Send a request for a new event table to the WTA server. The request must include the http header "time of last modification", indicating the date when the current event table content was modified.

3. Request the well known resource WTA Event table (conditional):

A request URL must be constructed using the well known resource name "wtatable.wtae" and the domain name of the WTA server.

Example: http://www.operator.com/wtatable.wtae

4. Reply from the WTA Server (conditional):

The WTA server must reply with a new event table content or include a result code using http headers [WSP] indicating that the requested content is the same as already stored in the client.

5. Setting up the event table (conditional):

If the WTA server wants to remove any existing event table an empty event table must be sent with the reply to the client.

6. Acknowledge to the WTA server (conditional):

The outcome of the set-up procedure for the event table in the WTA client is reported back to the WTA server using a result PDU with the acknowledge for the reply [WSP].

7. Conclusion of the start-up procedure (conditional):

The following rules must be applied when determining the state of the WTA client:

1. Acknowledge "success" - The acknowledge indicating success means that there is now a new event table in the WTA client ready for service.
2. Acknowledge "failure" - The acknowledge indicating failure means that the state of the event table in the WTA client is the same as indicated by the request from the WTA client for a new event table, i.e. existence of the "time of last modification" information.
3. Acknowledge lost - If an acknowledge to the reply with a new event table can not be received by the WTA server then the same rules applies as if an acknowledge with a failure was received.
4. The WTA server must assume the mode of the WTA client, server centric or client centric mode, based on the reported WTA user-agent characteristics [WAE]. See additional information on the WTA client mode in section [xx.yy].

11.2 Termination of a WTA session

The termination process makes it possible for an WTA server to gracefully detect that a WTA client has terminated the WTA session to the WTA server.

The following rules apply for determining the that the WTA client has been terminated:

1. The WTA server receives an indication from the WAP gateway that the WTA session has been terminated by the WTA client.
2. A WTA server that can not reach a WTA client due to non-existent session must assume the same state as if the WTA session would have been terminated by the WTA client.

11.3 Reliable Content Push

A WTA server can push content, decks and events, to the client's WTA port. The push services are provided by the WSP layer [WSP]. The reliable WSP service is generally referred to as acknowledged content in the WTA specification.

The WTA port is used for the reception of pushed content. The WTA server have the option to request an acknowledge message that the client actually received the pushed content. In order, for the WTA Server, to determine how to proceed when there is no user response the WTA server has to implement its own timeout handling. The WTA client must only receive one push at a time, which also has to be acknowledged back to the WTA server.

12 WTA Reference Information

This section defines the content format used to represent the WTA event and the WTA event table. WTA Event is an application of [XML] version 1.0.

12.1 Document Identifiers

Ed: these identifiers have not yet been registered with the IANA or ISO 9070 Registrar

12.1.1 SGML Public Identifier

```
--//WAPFORUM//DTD WTA 1.0//EN
```

12.1.2 WTA Event Media Type

Textual form:

```
text/x-wap.wtae
```

Tokenised form:

```
application/x-wap.wtaec
```

Ed: these types are not yet registered with the IANA and are consequently *experimental* media types.

12.2 Document Type Definition (DTD)

```
<!ELEMENT WTAI ( ( EVENTTABLE )+ | ( EVENT ) ) >

<!ELEMENT EVENTTABLE ( TYPE | URL | VAR )* >
<!ELEMENT EVENT ( TYPE | VAR | VALUE )* >

<!ELEMENT TYPE (#PCDATA) > <!-- Event type -->
<!ELEMENT URL (#PCDATA) > <!-- Event Table URL -->

<!-- Variable declaration -->
<!ELEMENT VAR EMPTY>
<!ATTLIST VAR
  NAME ID #REQUIRED
  >

<!-- Variable value -->
<!ELEMENT VALUE EMPTY>
<!ATTLIST VALUE
  NAME IDREF #REQUIRED
  CONTENT CDATA #REQUIRED
  >
```

13 WTA Event Binary Encoding

The WTA Event is encoded using a compact binary representation. This content format is based upon the WAP Binary XML Content Format [WBXML].

13.1 Extension Tokens

13.1.1 Global Extension Tokens

The [WBXML] global extension tokens are used to represent WTA event table variables. Same rules apply for WTA Event variables as for WML variables [WML].

13.1.2 Tag Tokens

WTA Event defines a set of single-byte tokens corresponding to the tags defined in the DTD. All of these tokens are defined within code page one.

13.1.3 Attribute Tokens

WTA Event defines a set of single-byte tokens corresponding to the attribute names and values defined in the DTD. All of these tokens are defined within code page one.

13.2 Encoding Semantics

13.2.1 Encoding Variables

All variable references must be converted to variable reference tokens (e.g. EXT_I_0).

13.2.2 Document Validation

XML document validation (see [XML]) should occur during the process of tokenising a WTA event, and must be based on the DOCTYPE declared in the WTA event. The tokenization process should notify the user of any well-formedness or validity errors detected in the source deck.

13.3 Numeric Constants

13.3.1 Tag Tokens

The following token codes represent tags in code page zero (0). All numbers are in hexadecimal.

Note: token assignments may change before final publication.

Table 3. Tag tokens

<u>Tag Name</u>	<u>Token</u>
EVENT	5
EVENTTABLE	6
TYPE	7
URL	8
WTAI	9

13.3.2 Attribute Start Tokens

The following token codes represent the start of an attribute in code page zero (0). All numbers are in hexadecimal.

Note: token assignments may change before final publication.

Table 4. Attribute start tokens	<u>Attribute Name</u>	<u>Attribute Value Prefix</u>	<u>Token</u>
	NAME		5
	VALUE		6

13.3.3 Attribute Value Tokens

No additional attribute values have been defined for WTA events. Same rule apply as for WML [WML].

13.4 WTA Encoding Examples

13.4.1 WTA Event

The following is an example of a tokenised WTA event. It demonstrates variable encoding, attribute encoding, and the use of the string table. Source WTA event:

```
<WTAI>
  <EVENT>
    <TYPE>cc/ic</TYPE>
    <VALUE CONTENT="1" />
    <VALUE CONTENT="6505551212" />
  </EVENT>
</WTAI>
```

Tokenised form (numbers in hex) follows. This example only uses inline strings, and assumes that the character encoding uses a NULL terminated string format. It also assumes that the character encoding is UTF-8:

Note: encoded example will be added later before final publication.

In an expanded and annotated form:

Table 5. Example tokenised WTA event

<u>Token Stream</u>	<u>Description</u>
00	WBXML Version number
01	END (of WTAI element)

13.4.2 WTA Event Table

The following is an example of a tokenised WTA event table. It demonstrates variable encoding, attribute encoding, and the use of the string table. Source WTA event table:

```
<?XML version="1.0" ?>
<!DOCTYPE WTAI SYSTEM "wtai.dtd">
<WTAI>
  <EVENTTABLE>
    <TYPE>cc/ic</TYPE>
    <URL>http://foo.com/call</URL>
    <VAR NAME="id"/>
    <VAR NAME="callerID"/>
  </EVENTTABLE>

  <EVENTTABLE>
    <TYPE>cc/if</TYPE>
    <URL>http://foo.com/flash</URL>
    <VAR NAME="string"/>
  </EVENTTABLE>
</WTAI>
```

Tokenised form (numbers in hex) follows. This example only uses inline strings, and assumes that the character encoding uses a NULL terminated string format. It also assumes that the character encoding is UTF-8:

Note: *encoded example will be added later* before final publication.

In an expanded and annotated form:

Table 6. Example tokenised WTA event

<u>Token Stream</u>	<u>Description</u>
00	WBXML Version number
01	END (of WML element)

14 WTA Examples

There are probably several ways to create these services and the use cases must consequently only be considered as examples, not necessarily the easiest or the foolproof ways to handle them.

14.1 Provisioning

The example assumes a new WAP MS without any preloaded WTA content. The question would then be how to activate the WTA features. Below is the sequence for the WAP transactions between the MS and the WTA Server

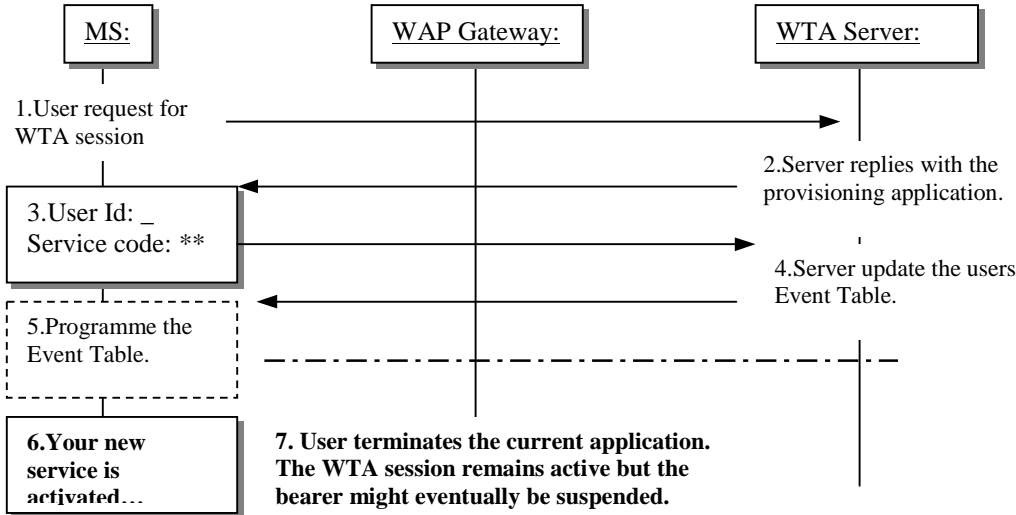


Figure 2, WTA Service “Provisioning”

Walkthrough the Service “Provisioning”:

- 1) The network operator may have informed user, about the location of the “WTA” WAP Gateway². Starting the WTA user-agent, from the user perspective, also means that the WAP stack is initialised. The “preferred” Gateway is forwarded to WSP and consequently WTP. The user requests a new WTA session.
- 2) The WTA server replies with the “Provisioning Application”.
- 3) The user enters his subscription id and “Requested Service Code”.
- 4) The server checks the user and then sends down the updated appropriate event table.
- 5) The WTA client uses the downloaded content to “program” the new event table.
- 6) After the successful operation, the user is notified that the service is activated.
- 7) Assuming the handset is powered up and WAP is enabled the user is now ready for WTA.

² The information on the location of the WAP gateway or the WTA Server either is manually entered into the device, or can be part of the configuration.

14.2 Incoming Call

“Incoming Call” WTA use case is an example how an application could be deployed using features in the client centric mode of the WTA architecture. The example assumes that the content, in the form of WML and WMLScript already is stored in the WAP client. The example shows how a “generic” application WTA event activates the handling of a call application and gives the called party value added services like pre-call options in the form of a menu. The normal mobile network call handling must be override in order to handle the call and intercept the standard call handling in the mobile phone. Below is the outline of the sample service network.

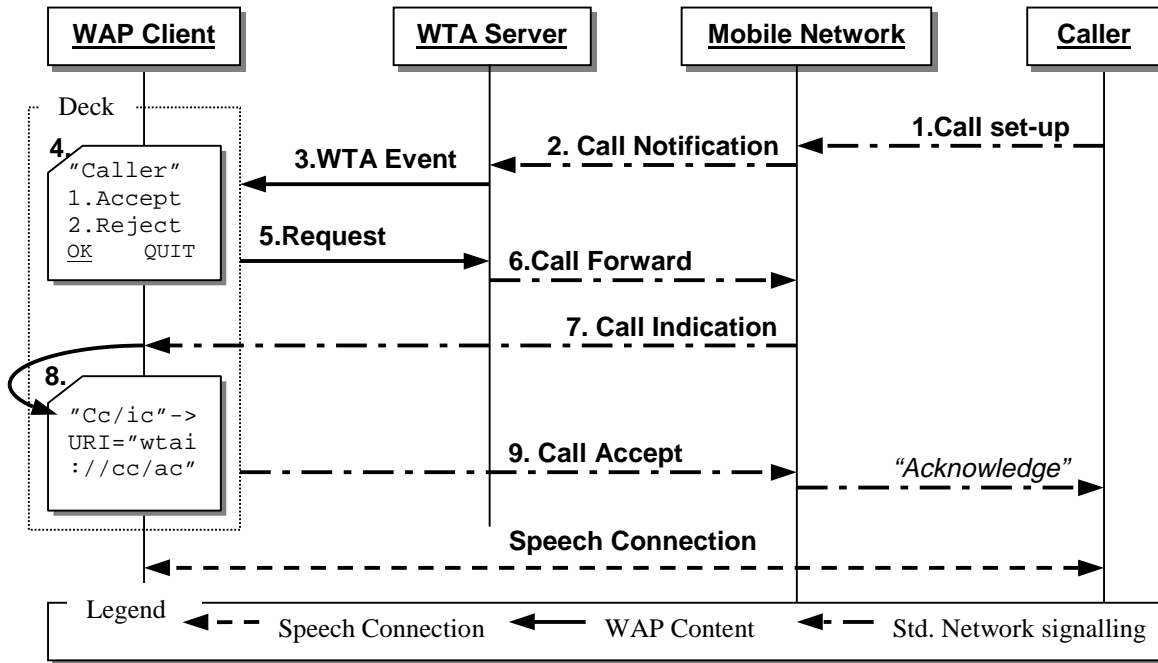


Figure 3, WTA "Incoming Call"

Walkthrough of the “Incoming Call” application:

1. A “Caller” intends to call the WAP client end-user. The mobile network, configured to divert the calls to another entity, temporarily holds the Call set-up request from the “Caller.
2. The mobile network sends a call notification to the WTA server notifying about the pending call.
3. The WTA server pushes down a WTA event that is detected by the call control content.
4. The WAP Client end user is presented with a menu that tells him about the incoming call and the different options for how to proceed (The WTAI “logical Indication” function alerts the end-user)
5. The WAP client end-user decides to accept the call. A request for the “accept call” service is sent to the WTA Server.
6. The WTA server instructs the mobile network to forward the held call to the WAP client.
7. The mobile network allocates the traffic channel and sends a “call indication” the WAP Client.
8. An “Incoming Call” WTA Event is generated locally as a result of the network event “call indication”.
9. As the content detects the pre configured WTA event the, WTA enabled WML/WMLScript, content³ answers the call using WTAI functions.

The example is concluded with the mobile network establishing the actual call between the two parts.

³The standard Mobile Phone MMI does not intervene or generate the standard “Incoming Call” notification.

14.3 Voice Mail

Voice Mail interaction between the WTA server and WTA client gives an example of an application, that starts up as a standard “browsing” use case and, with the help from the WTA server, eventually invokes a voice connection.

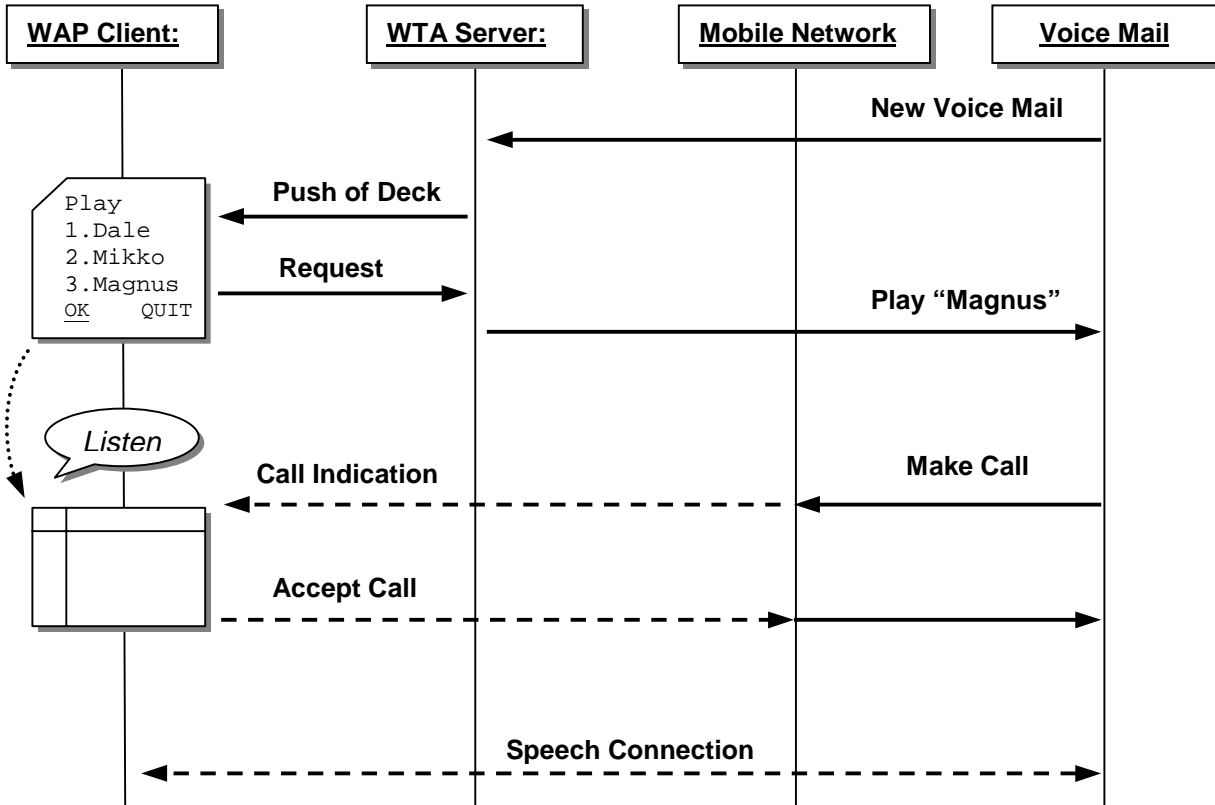


Figure 8, Voice Mail Sequence

Walkthrough:

1. In the beginning there is a new voicemail in the voicemail system.
2. The voicemail system informs the WTA server about the voicemails.
3. WTA server creates a deck, which is pushed to the client.
4. The client shows a list of voicemails to the user and the user selects a voicemail to listen for.
5. A request is sent back to the WTA server.
6. At the same time a new card is shown to the user (“Listen”) and the network event, “incoming call” is mapped using a WTA event binding in order to be answered automatically.
7. The WTA server sends information back to the voicemail system about which voice mail to play.
8. The voice mail system makes the call to the WTA client.
9. Since there is an event mapping the incoming call is answered automatically and therefore the client can accept the call without the need for user interaction.
10. When the speech connection is established the voice mail system starts playing the voice mail and the user can listen.

