

# WAP WTAI (TETRA)

WAP-246-WTAITETRA-t  
Prototype Version 16-May-2001

---

for

## Wireless Application Protocol Wireless Telephony Application Interface Specification TETRA Specific Addendum

This document is a work in process and is not an approved WAP Forum™ specification. This document is subject to revision or removal without notice. No part of this document may be used to claim conformance or interoperability with the WAP Forum specifications.

© 2001, Wireless Application Protocol Forum, Ltd. All Rights Reserved. Terms and conditions of use are available from the WAP Forum™ Web site (<http://www.wapforum.org/docs/copyright.htm>).

© Wireless Application Protocol Forum Ltd. 2001.  
All rights reserved

© 2001, Wireless Application Protocol Forum, Ltd. All rights reserved.  
Terms and conditions of use are available from the WAP Forum™ Web site at  
<http://www.wapforum.org/what/copyright.htm>.

You may use this document or any part of the document for internal or educational purposes only, provided you do not modify, edit or take out of context the information in this document in any manner. You may not use this document in any other manner without the prior written permission of the WAP Forum™. The WAP Forum authorises you to copy this document, provided that you retain all copyright and other proprietary notices contained in the original materials on any copies of the materials and that you comply strictly with these terms. This copyright permission does not constitute an endorsement of the products or services offered by you.

The WAP Forum™ assumes no responsibility for errors or omissions in this document. In no event shall the WAP Forum be liable for any special, indirect or consequential damages or any damages whatsoever arising out of or in connection with the use of this information.

This document is not a WAP Forum™ specification, is not endorsed by the WAP Forum and is informative only. This document is subject to revision or removal without notice. No part of this document may be used to claim conformance or interoperability with the WAP Forum specifications.

WAP Forum™ members have agreed to use reasonable endeavors to disclose in a timely manner to the WAP Forum the existence of all intellectual property rights (IPR's) essential to the present document. The members do not have an obligation to conduct IPR searches. This information is publicly available to members and non-members of the WAP Forum and may be found on the "WAP IPR Declarations" list at <http://www.wapforum.org/what/ipr.htm>. Essential IPR is available for license on the basis set out in the schedule to the WAP Forum Application Form.

No representations or warranties (whether express or implied) are made by the WAP Forum™ or any WAP Forum member or its affiliates regarding any of the IPR's represented on this list, including but not limited to the accuracy, completeness, validity or relevance of the information or whether or not such rights are essential or non-essential.

This document is available online in PDF format at <http://www.wapforum.org/>.

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

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

Document History	
WAP-246-WTAITETRA-20010516-t	Current

---

# Contents

<a href="#">CONTENTS</a>	3
<a href="#">1. SCOPE</a>	5
<a href="#">2. DOCUMENTS STATUS</a>	6
<a href="#">2.1 COPYRIGHT NOTICE</a>	6
<a href="#">2.2 ERRATA</a>	6
<a href="#">2.3 COMMENTS</a>	6
<a href="#">3. REFERENCES</a>	7
<a href="#">3.1 NORMATIVE REFERENCES</a>	7
<a href="#">3.2 INFORMATIVE REFERENCES</a>	7
<a href="#">4. DEFINITIONS AND ABBREVIATIONS</a>	8
<a href="#">4.1 DEFINITIONS</a>	8
<a href="#">4.2 ABBREVIATIONS</a>	8
<a href="#">5. TETRA BACKGROUND</a>	10
<a href="#">5.1 TETRA VOICE CALL MODEL</a>	10
<a href="#">5.1.1 TETRA Voice Call States</a>	10
<a href="#">5.1.2 TETRA Voice Call Information</a>	14
<a href="#">5.1.3 TETRA Addressing</a>	14
<a href="#">5.1.4 TETRA Addresses Syntax Definitions</a>	14
<a href="#">5.2 TETRA NETWORK MESSAGE MODEL</a>	15
<a href="#">5.2.1 TETRA Network Message State</a>	15
<a href="#">5.2.2 TETRA Network Message Features</a>	15
<a href="#">5.2.3 Network Message Identifier</a>	16
<a href="#">5.2.4 Network Message Information</a>	16
<a href="#">6. SPECIAL BEHAVIOUR OF NETWORK COMMON WTAI</a>	18
<a href="#">6.1 WTA EVENTS</a>	18
<a href="#">6.1.1 wtaev-cc/cl</a>	18
<a href="#">7. NETWORK SPECIFIC WTAI - TETRA</a>	19
<a href="#">7.1 VOICE CALL EVENTS</a>	19
<a href="#">7.1.1 wtaev-tetra/tg</a>	19
<a href="#">7.1.2 wtaev-tetra/rg</a>	19
<a href="#">7.1.3 wtaev-tetra/cq</a>	19
<a href="#">7.1.4 wtaev-tetra/re</a>	19
<a href="#">7.1.5 wtaev-tetra/ic</a>	20
<a href="#">7.1.6 wtaev-tetra/da</a>	20
<a href="#">7.1.7 wtaev-tetra/ua</a>	20
<a href="#">7.2 WMLSCRIPT FUNCTIONS</a>	20
<a href="#">7.2.1 WTATETRA.makeCall</a>	21
<a href="#">7.2.2 WTATETRA.sdsSend</a>	22
<a href="#">7.2.3 WTAITETRA.sdsSendAck</a>	23
<a href="#">APPENDIX A. WMLSCRIPT LIBRARY FUNCTIONS</a>	24
<a href="#">APPENDIX B. STATIC CONFORMANCE REQUIREMENTS</a>	25

<a href="#">B.1 CLIENT FEATURES</a> .....	25
<a href="#">B1.1 TETRA Network Model</a> .....	25
<a href="#">B1.2 WTA Events</a> .....	25
<a href="#">B1.3 WMLScript Functions</a> .....	25
<a href="#">B1.4 WMLScript Bytecode Interpreter Capabilities</a> .....	26
<a href="#">B2. SERVER FEATURES</a> .....	26
<a href="#">B2.1 WMLScript Encoder Capabilities</a> .....	26
<a href="#">APPENDIX C. SPECIFICATION-TRACK DOCUMENT HISTORY</a> .....	27

---

# 1. Scope

Wireless Application Protocol (WAP) is a result of continuous work to define an industry wide specification for developing applications that operate over wireless communication networks. The scope for the WAP Forum is to define a set of specifications 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*" [WAPARCH].

This document is an addendum to the *Wireless Telephony Application Interface* [WTAI]. While WTAI defines an API that is valid for all supported types of mobile networks, this document outlines functions that are specific to networks using TETRA technology.

---

## 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, 2000. Terms and conditions of use are available from the Wireless Application Protocol Forum Ltd. web site at <http://www.wapforum.org/docs/copyright.htm>.

### 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

- [EBNF] W3C, "REC-xml-19980210: Extensible Mark-up Language (xml) 1.0", February 10, 1998
- [ETS 300 392-1] European Telecommunication Standard. Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 1: General Network Design; 2000.
- [ETS 300 392-2] European Telecommunication Standard. Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI); October 2000.
- [GENFORM] WAP-188, "WAP General Formats Document", WAP Forum. URL: <http://www.wapforum.org/>
- [RFC2119] "Key words for use in RFCs to Indicate Requirement Levels", S. Bradner, March 1997. URL: <http://www.ietf.org/rfc/rfc2119.txt>
- [RFC2396] "Uniform Resource Identifiers (URI): Generic Syntax." T. Berners-Lee, R. Fielding, L. Masinter, et al., August 1998. URL: <http://www.ietf.org/rfc/rfc2396.txt>
- [WMLScript] "WMLScript Language Specification", WAP Forum, June 2000. URL: <http://www.wapforum.org/>
- [WTA] WAP-169, "Wireless Telephony Application Specification", WAP Forum. URL: <http://www.wapforum.org/>
- [WTAI] WAP-170, "Wireless Telephony Application Interface Specification", WAP Forum. URL: <http://www.wapforum.org/>

### 3.2 Informative references

- [WAPARCH] "Wireless Application Protocol Architecture Specification", WAP Forum, 30-Apr-1998. URL: <http://www.wapforum.org/>

---

## 4. Definitions and Abbreviations

The following section describes definitions and abbreviations common to this document.

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 in [RFC2119].

### 4.1 Definitions

The following are terms and conventions used throughout this specification.

**Hook Signalling** – signalling procedure which includes an alerting process to the called party. An explicit response indicating the called party answered is waited, before the call can be set up.

**Direct Signalling** – signalling procedure where immediate communication with the called party can take place, without the alerting process and without explicit response indicating the called party has answered.

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

### 4.2 Abbreviations

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

<b>API</b>	Application Programming Interface
<b>EBNF</b>	Extended Backus-Naur Form
<b>MCC</b>	Mobile Country Code
<b>GSSI</b>	Group SSI
<b>GTSI</b>	Group TSI
<b>ITSI</b>	Individual TSI
<b>MNC</b>	Mobile Network Code
<b>MNI</b>	Mobile Network Identity
<b>MR</b>	Message Reference
<b>PSTN</b>	Public Switched Telephone Network
<b>PTT</b>	Push To Talk button/switch.
<b>RFC</b>	Request For Comments
<b>SDS</b>	Short Data Service
<b>SNA</b>	Short umber Address
<b>SSI</b>	Short Subscriber Identity
<b>TETRA</b>	Terrestrial Trunked Radio
<b>TSI</b>	TETRA Subscriber Identity
<b>URI</b>	Uniform Resource Identifier [RFC2396]
<b>WAP</b>	Wireless Application Protocol [WAPARCH]
<b>WTA</b>	Wireless Telephony Applications [WTA]
<b>WTAI</b>	Wireless Telephony Applications Interface [WTAI]



---

## 5. TETRA Background

### 5.1 TETRA Voice Call Model

The WTA TETRA voice call model is an extension of the network-common WTA voice call model described in [WTAI].

#### 5.1.1 TETRA Voice Call States

The following call models for TETRA are specified in this document:

- Half-duplex, individual and group calls, direct signalling (without hook signalling).
- Half duplex, individual call, non-direct signalling (with hook signalling).
- Full duplex, individual call, direct signalling (without hook signalling).

All other TETRA call types, such as phone calls, adhere to the call models specified in [WTAI], but they have optional extended states..

The outgoing and incoming call model for the above TETRA call types are shown in Figure 1, Figure 2, Figure 3 and Figure 4. They represent the lifetime of one call and show the call states and the events that result in state transitions. A voice call may stay in a particular state for as long as allowed by the system (i.e. until certain system timers expire).

In TETRA, in half-duplex call, a call request can be initiated while the other party is talking (usually by pressing PTT). However, for simplicity, this is not possible in the call model described in this document.

TETRA WTA implementations must generate WTA events according to the models specified here. WTA implementations will generally rely on the underlying network signalling such that:

- Network events correspond to zero or more WTA events
- The WTA implementations can support all these call models without maintaining call state
- No WTA events need to be generated without an underlying network event

When initiating a call in the no hook (direct) signalling case, in the event that the subscriber is unavailable, for any particular reason, the appropriate error code will be returned.

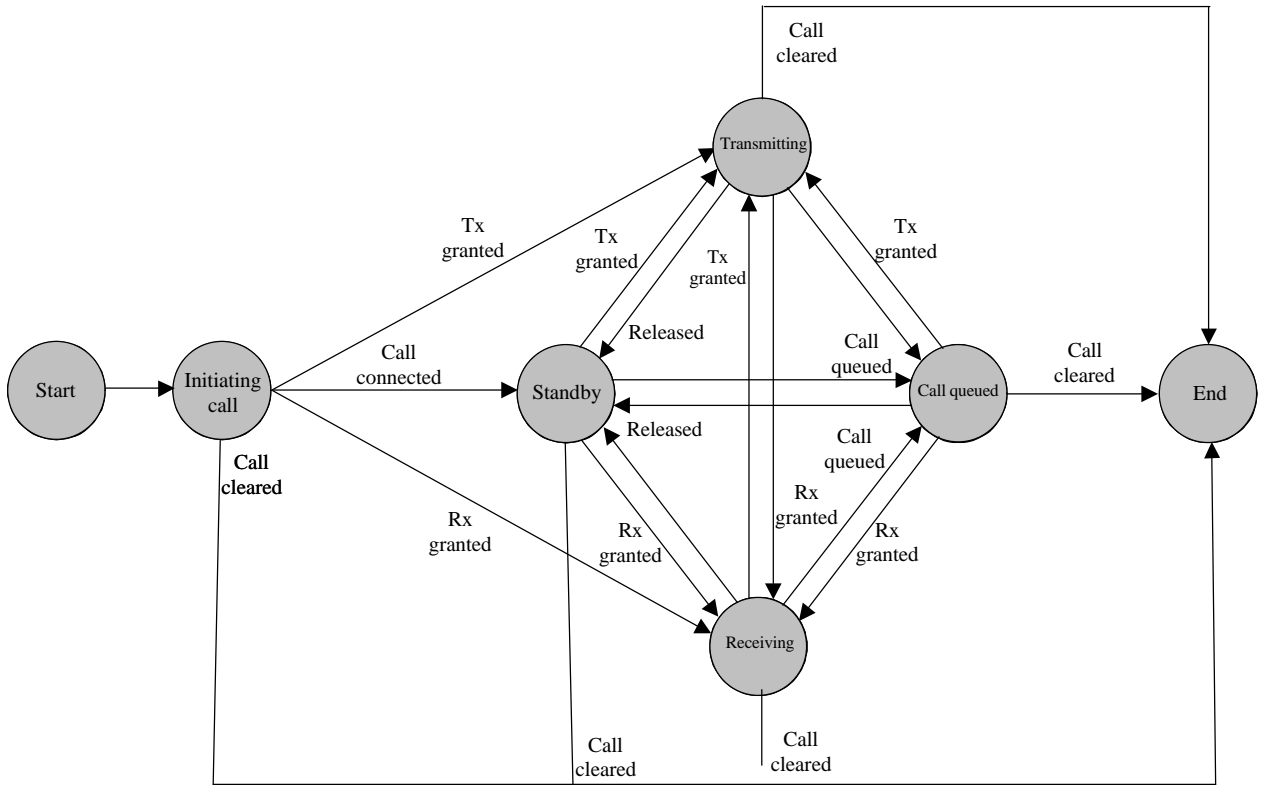


Figure 1. Outgoing half-duplex, group or individual call, direct signalling (without hook signalling)

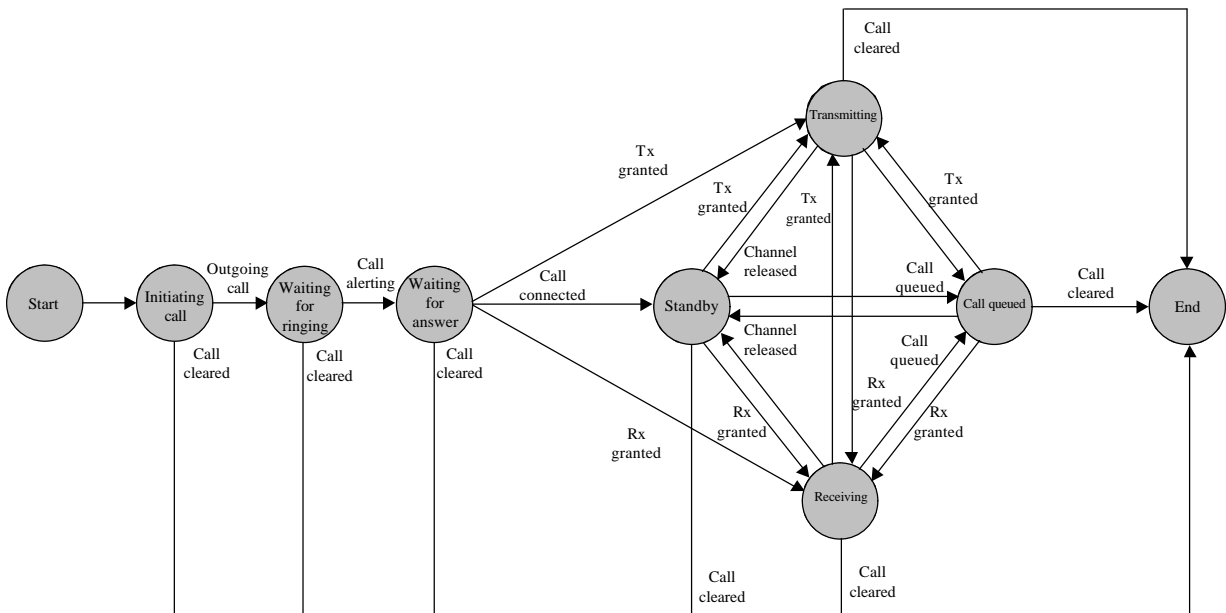


Figure 2. Outgoing half-duplex individual call, non direct signalling (with hook signalling)

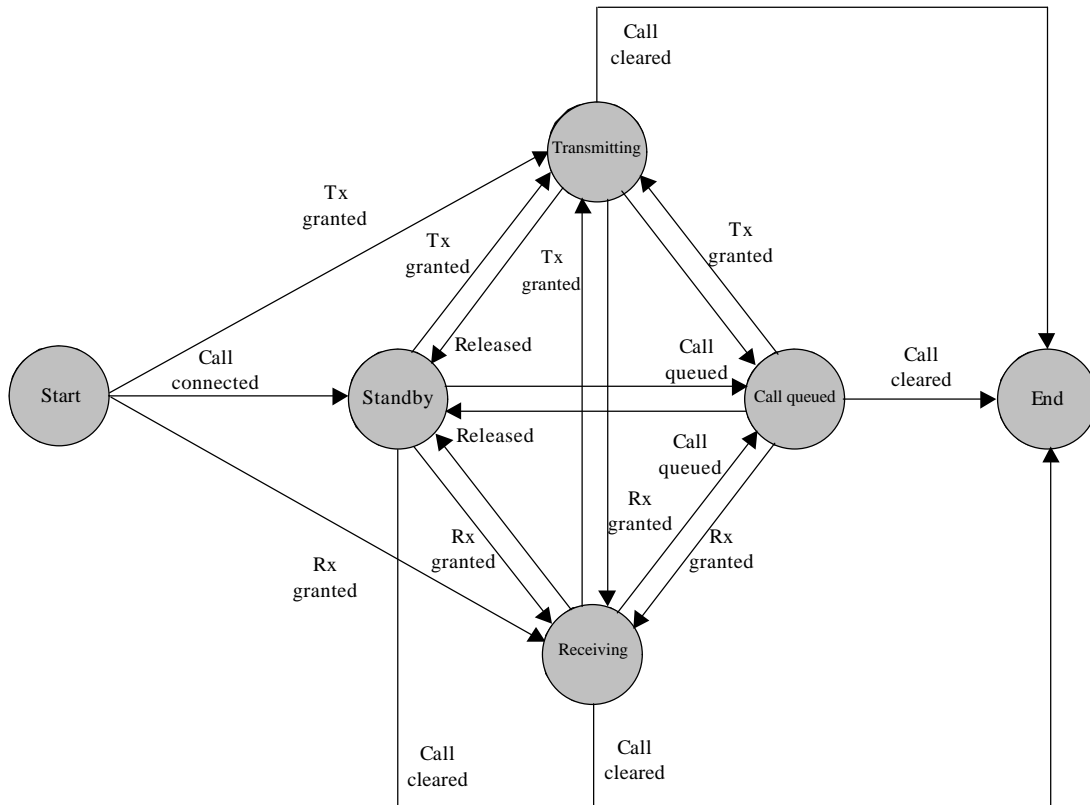


Figure 3. Incoming half duplex, individual or group call, direct signalling (without hook signalling)

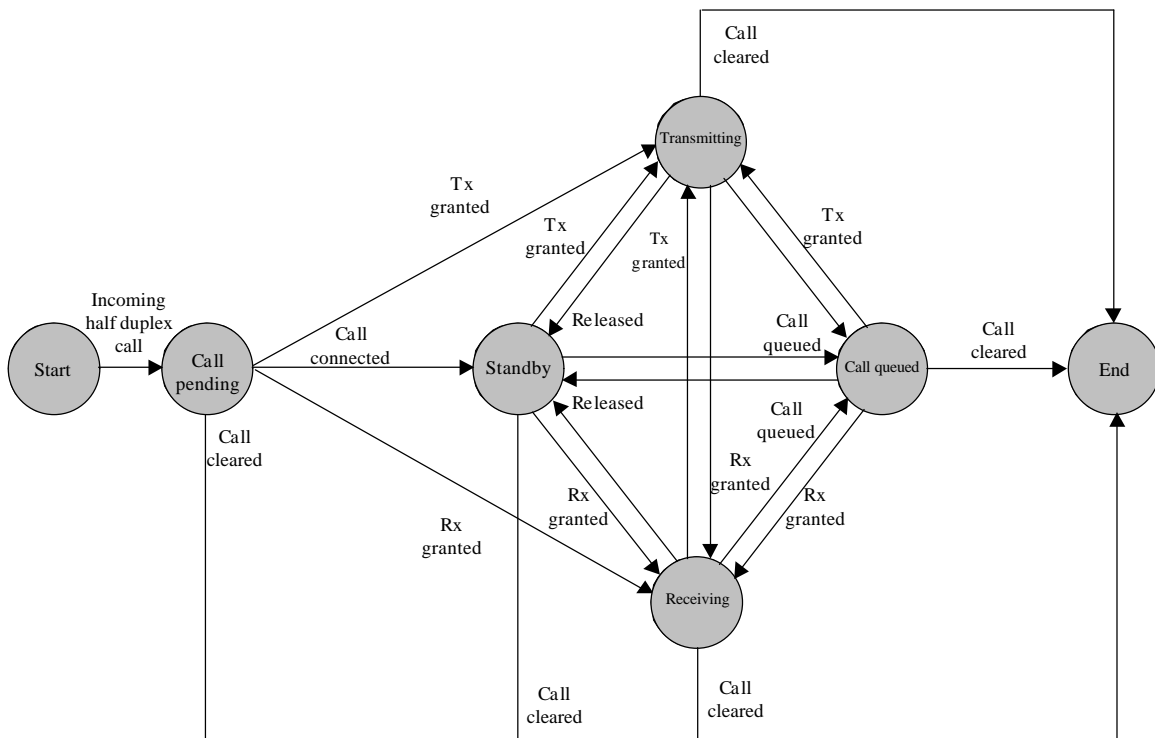


Figure 4. Incoming half duplex, individual or group call, non direct signalling (with hook signalling)

## 5.1.2 TETRA Voice Call Information

A WTA user agent that implements the TETRA library provides access to information about each TETRA voice call state. Each information field has a name and a value. A field value may be retrieved using its field name.

"TETRAstatus" integer indicating the recent state of the call in the TETRA Outgoing Call model diagrams. Note that this field may not be completely accurate or up to date since the call state may have changed at anytime. The integer value must be one of the following decimal values:

- 1 = the TETRA call is in the "initiating call" state
- 2 = the TETRA call is in the "waiting for ringing" state
- 3 = the TETRA call is in the "waiting for answer" state
- 4 = the TETRA call is in the "transmitting" state
- 5 = the TETRA call is in the "receiving" state
- 6 = the TETRA call is in the "standby" state
- 7 = the TETRA call is in the "call queued" state
- 8 = the TETRA call is in the "end" state

## 5.1.3 TETRA Addressing

The different addresses used in TETRA are defined in [ETS 300 392-1]. TETRA has two main types of addresses:

- Phone number (MS-ISDN) which is used for phone calls as in other cellular systems.
- TETRA specific ID which allows to call other TETRA devices, either individually or a group.

The address types that are used in this document are:

- **SNA** – Short Number Address. 8 bits. Used for short dialling to numbers pre-stored in the system.
- **SSI** – Short Subscriber Number. 24 bits. TETRA specific ID. Can be either ISSI (Individual SSI) which is used by a specific device or GSSI (Group SSI) which can be used by a group of devices.
- **TSI** – TETRA Subscriber Identity – 48 bits. Used for inter-system TETRA calls. 24 bits are ISSI and the additional 24 bits are the MNI, which is MCC and MNC. TSI Can be either ITSI for individual addressing or GTSI for group addressing.
- **Phone-number** – (MS-ISDN, Mobile Station ISDN number, in TETRA terminology). Regular phone number.

Despite the different representations in TETRA as bit strings, for the WTAI functions, all the numbers are presented as text strings. Additional call type parameter defines their meaning.

## 5.1.4 TETRA Addresses Syntax Definitions

This section defines a general format for the specific TETRA address used in this document (SNA, SSI, TSI). Phone number syntax is defined in [GENFORMAT]. The format is given using EBNF [EBNF]. EBNF reserves the hyphen character and does not capitalize non-terminal variables by convention. These rules are followed in the syntax definitions below. Outside of this specification, however, one may refer to variables using capital letters and with hyphens instead of underscores unambiguously, as these devices are not used to distinguish between variables in the definitions.

```

Tetra address string ::= ( SNA | SSI | TSI )
SNA                   ::= [0-2]? DIGIT DIGIT
TSI                   ::= "+" MNI "#" SSI
SSI                   ::= DIGIT+
MNI                   ::= DIGIT+

```

DIGIT ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

## 5.2 TETRA Network Message Model

### 5.2.1 TETRA Network Message State

TETRA network text messages are called Short Message Service (SMS) messages. The WTA network message model in TETRA for incoming messages is identical to Network Common WTAI Message model for incoming messages (See [WTAI]). For outgoing messages, the message model is illustrated in Figure 1 below:

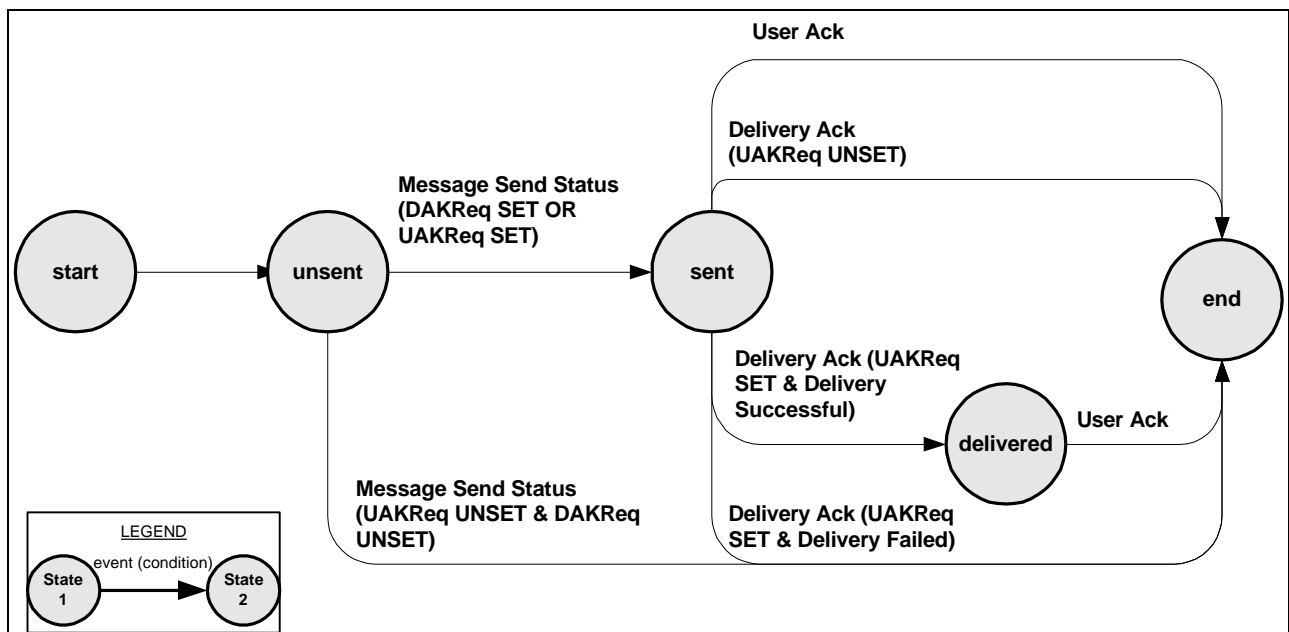


Figure 1. TETRA Outgoing Message Model

### 5.2.2 TETRA Network Message Features

The TETRA standard allows two types of acknowledgements for text messages. When a user sends a text message, it is possible to request the following delivery acknowledgements:

- SDS message received report. For compatibility, this report is called in this document Delivery Acknowledge.
- SDS message consumed report. For compatibility, the report is called in this document User Acknowledge.

The SDS message received report (delivery ack) confirms that the destination device has successfully received the message. The SDS message consumed (user ack) report confirms that the message has been consumed by an application at the destination, i.e the user has acknowledged receiving the message manually or automatically. A user can request both types delivery reports. It is implicitly assumed that delivery acknowledgements are received as normal SDS messages.

If the "delivery\_ack" field received in the incoming SDS message indicates that the delivery acknowledgement (received report) is requested, the service should send Delivery Acknowledgement.

If the "user\_ack" field received in the incoming SDS message indicates that the user acknowledgement (consumed report) is requested, the service may indicate the request to the user. If the request was indicated to the user, then when

the user acknowledges the message, service should send User Acknowledgement. If the request was not indicated to the user, then the service should send User Acknowledgement when the user reads the message.

Sending the acknowledgements is done by the WMLScript function `WTAITETRA.sendAck` invoked to send the acknowledgement message to the sender of the message. (See section 7.2.3 for details of `WTAITETRA.sendAck` function).

**Note:** The SDS Delivery Acknowledgment and the SDS User Acknowledgment are generated only if requested by the sender of the original SDS message. If the SDS message was sent using `WTANetText.send` function (see [WTAI]), acknowledge options should be set based on pre-defined user settings. User settings/preferences are set in an implementation specific manner.

### 5.2.3 Network Message Identifier

A `MESSAGE_ID` field is included in every outgoing or incoming TETRA SDS message (in TETRA terms, the message ID is called MR – Message reference). The value of the `MESSAGE_ID` field is generated as defined for Message References in [ETS 300 392-2] and is used in the end to end acknowledgements. TETRA implementations MUST use the `MESSAGE_ID` field value to associate SDS Delivery Acknowledgments or SDS User Acknowledgments with the acknowledged SDS messages. The `MESSAGE_ID` should be used as described below

- If an SDS Delivery Acknowledgment or an SDS User Acknowledgment is requested in an outgoing message, WTA-TETRA implementations MUST associate the value of the “`msgHandle`” assigned to the outgoing WTA network message (see [WTAI]) with the value of the `MESSAGE_ID` field in the sent TETRA SDS message. When the SDS Delivery Acknowledgment or the SDS User Acknowledgment message is received, WTA-TETRA implementations MUST associate the `MESSAGE_ID` field that is included in the Acknowledgment message with the “`origMsgHandle`” field. A WTA service uses the value of the “`origMsgHandle`” field in the received acknowledgement to determine and retrieve information about the acknowledged SDS message.
- If an SDS User Acknowledgment is requested in an incoming message, WTA-TETRA implementations MUST associate the value of the `MESSAGE_ID` field with the “`msgHandle`” of the incoming WTA network message (see [WTAI]). When the WMLScript function `WTAITETRA.sdsSendAck` is invoked to send a User Acknowledgment message, the “`msgHandle`” parameter value MUST be used to retrieve the `MESSAGE_ID` of the SDS message to be acknowledged.

The way the `MESSAGE_ID` (and possibly other SDS parameters) is associated with the “`msgHandle`” or “`origMsgHandle`” is implementation specific.

### 5.2.4 Network Message Information

The WTA user agent provides access to specific information about each network message. Each information field has name and a value. A field value may be retrieved using its name. All the information fields defined in [WTAI] for network messages are available in TETRA network messages. However, depending on the type of the network message, extra information may be available to the service as described below.

#### SDS Message

The SDS message contains all the fields defined in [WTAI] for network messages. In addition, the following fields may be available for each incoming and outgoing SDS message:

- “`delivery_ack`” boolean indicating whether the SDS Delivery (Received) Acknowledgement is requested or not requested for this message. This field is set to ‘`true`’ if the SDS Delivery Acknowledgment is requested. Otherwise, it is set to ‘`false`’.
- “`user_ack`” boolean indicating whether the SDS User (Consumed) Acknowledgement is requested or not requested for this message. This field is set to ‘`true`’ if the SDS User Acknowledgment is requested. Otherwise, it is set to ‘`false`’.

## SDS Delivery Acknowledgement

All the fields defined in [WTAI] for network messages must be included in the SDS Delivery Acknowledgment message.

In addition to the fields defined in [WTAI], The SDS Delivery acknowledgement must include the following fields:

*“origMsgHandle”* handle identifying the message that has been acknowledged. The value of this field is the same as the value of the *“msgHandle”* of the SDS message being acknowledged.

The *“text”* field in the SDS acknowledgement contains text added by the MC as described below:

*“text”* string containing the body of the SDS Delivery acknowledgement message. The body of the message will be one of the following:

- An empty string indicating that the recipients of the SDS message has confirmed the reception of the message.
- An error message, in case the delivery is unsuccessful or an error is reported.

## SDS User Acknowledgement

In addition to the fields defined in [WTAI] for network messages, the SDS User Acknowledgement must include the following fields:

*“origMsgHandle”* handle identifying the SDS message the end recipient acknowledged. The value of this field is the same as the value of the *“msgHandle”* of the SDS message being acknowledged.

---

## 6. Special Behaviour of Network Common WTAI

This section describes the changes to the semantics or behaviour of the network common portion of WTA.

### 6.1 WTA Events

These network common events behave differently or have additional characteristics when used in a WTA device implementing the TETRA library.

#### 6.1.1 wtaev-cc/cl

Description:

In addition to the description specified in [WTAI], this event can be used to indicate that the TETRA call has ended. The *result* parameter contains a description of why the call was cleared. It must be one of the values defined in [WTAI], section 8.1.2, or:

- "200" Timer expired
- "201" Called party not reachable
- "202" User requested disconnect
- "203" Acknowledged service not completed
- "204" SwMI requested disconnection
- "205" Pre-emptive use of resource
- "206" Called party does not support encryption
- "207" Congestion in infrastructure
- "208" Not allowed traffic case
- "209" Incompatible traffic case
- "210" Requested service not available
- "211" Invalid call identifier
- "212" Call rejected by called party
- "213" Loss of resources



---

## 7. Network Specific WTAI - TETRA

In addition to the WTA functions defined in [WTAI], TETRA networks also support the functions specified in this chapter. Since TETRA is the predecessor, the function library is named using that abbreviation.

### 7.1 Voice Call Events

These events are related to a TETRA device. The event parameters are all conveyed as strings.

#### 7.1.1 wtaev-tetra/tg

**Event Name:** Tx granted

**Event ID:** wtaev-tetra/tg

**Parameters:** *callHandle*, *callerId*

**Description:** Indicates a channel has been granted by the network for the user to start transmitting. The *callHandle* parameter contains the call handle for the TETRA call as defined in [WTAI].

The *callerId* parameter contains the called party's group or individual number, if available, as defined in section 5.1.4. If the receiving party's number cannot be determined, the *callerId* parameter must contain an empty string.

#### 7.1.2 wtaev-tetra/rg

**Event Name:** Rx granted

**Event ID:** wtaev-tetra/rg

**Parameters:** *callHandle*, *callerId*

**Description:** Indicates a channel has been granted by the network for the user to start receiving. The *callHandle* parameter contains the call handle for the TETRA call as defined in [WTAI].

The *callerId* parameter contains the calling party's group or individual number, if available, as defined in section 5.1.4. If the transmitting party's number cannot be determined, the *callerId* parameter must contain an empty string.

#### 7.1.3 wtaev-tetra/cq

**Event Name:** Call queued

**Event ID:** wtaev-tetra/cq

**Parameters:** *callHandle*

**Description:** Indicates a call has been queued.

The *callHandle* parameter contains the call handle for the TETRA call as defined in [WTAI].

#### 7.1.4 wtaev-tetra/re

**Event Name:** Released

**Event ID:** wtaev-tetra/re

**Parameters:** *callHandle*

**Description:** Indicates a channel (or a request for a channel in the call queued state) has been released, and the call has returned to the standby state. The *callHandle* parameter contains the call handle for the released TETRA call as defined in [WTAI].

### 7.1.5 wtaev-tetra/ic

**Event Name:** Incoming half-duplex non-direct call

**Event ID:** wtaev-tetra/ic

**Parameters:** *callHandle*, *callerId*

**Description:** Indicates that there is an incoming half duplex voice call. This event is only generated for non direct signalling calls (with hook signalling).

The *callHandle* parameter contains the call handle for the incoming call as defined in [WTAI].

The *callerId* parameter contains the calling party's group or individual number, if available, as defined in section 5.1.4. If the calling party's number cannot be determined, the *callerId* parameter must contain an empty string.

### 7.1.6 wtaev-tetra/da

**Event Name:** DeliveryAck

**Event ID:** wtaev-tetra/da

**Parameters:** *msgHandle*

**Description:** Indicates the client has received a SDS User Acknowledgement (Received Report) message.

The *msgHandle* parameter contains the message handle for the received SDS Delivery Acknowledgement message as defined in [WTAI].

### 7.1.7 wtaev-tetra/ua

**Event Name:** UserAck

**Event ID:** wtaev-tetra/ua

**Parameters:** *msgHandle*

**Description:** Indicates the client has received a SDS User Acknowledgement (Consumed Report) message.

The *msgHandle* parameter contains the message handle for the received SDS User Acknowledgement message as defined in [WTAI].

## 7.2 WMLScript functions

**Name:** WTATETRA

**Library ID:** 522

**Description:** This library contains functions that are specific to WTA implementations for TETRA.

## 7.2.1 WTATETRA.makeCall

**Function:** `makeCall (calltype, simplex, call_priority, called_party_type_id, hook, number, mode)`

**Function ID:** 0

**Description:** Initiates a mobile originated call. This function can be used to initiate all TETRA call types. This function is non-blocking. Subsequent WTA events signal the call progress.

This function returns an empty `string` if successful or returns `invalid` if the function fails.

**Note:** Only full duplex TETRA calls can be initiated using the core WTAI library in [WTAI]. Half duplex TETRA calls must be initiated using the above function.

**Permission Types:** BLANKET, CONTEXT, SINGLE (see [WTAI]).

**Parameters:** `calltype = boolean`

0 = Individual

1 = Group

`simplex = boolean`

true = Simplex

false = Duplex

`call_priority = integer`

0 = Default

1 = Low

2 = Medium

3 = High

4 = Emergency

`called_party_type_id = integer`

0 = Short number address (SNA)

1 = Short subscriber identity (SSI)

2 = TETRA subscriber identity (TSI)

`hook = boolean`

true = Hook on/off signalling (individual call); and call acceptance signalling (group call)

false = No hook signalling (direct signalling)

`number = string`

`mode = boolean` (false=drop, true=keep) (See [WTAI] for the description of Voice Call Modes)

**Return Value:** `handle` or `invalid`.

**Associated Events:** `Outgoing call (wtaev-cc/oc)`

`Call cleared (wtaev-tetra/cc)`

`Rx granted (wtaev-tetra/rg)`

Tx granted (wtaev-tetra/tg)

**Exceptions:** If the number parameter is not a valid SNA, SSI or TSI as defined in 5.1.4, the function returns *invalid*.

**Example:** var temp = WTATETRA.makeCall (0, true, 0, 1, false, "5554367", true);

## 7.2.2 WTATETRA.sdsSend

**Function:** sdsSend (*message\_type*, *called\_party\_type\_id*, *message\_receipt*, *message\_consumed*, *address*, *text*)

**Function ID:** 1

**Description:** Send a short data service text message. This function is non-blocking. Subsequent WTA events signal the message transition progress.

The *called\_party\_type\_id* parameter indicates the type of address used.

The *delivery\_ack* parameter indicates whether a SDS received confirmation is required.

The *user\_ack* parameter indicates whether a SDS consumed confirmation is required.

**Permission Types:** BLANKET, CONTEXT, SINGLE (see [WTA]).

**Parameters:** *message\_type* = integer

0 = Status

1 = SDS type 1

2 = SDS type 2

3 = SDS type 3

4 = SDS type 4

*called\_party\_type\_id* = integer

0 = Short number address (SNA)

1 = Short subscriber identity (SSI)

2 = TETRA subscriber identity (TSI)

3 = Phone-Number

*delivery\_ack* = boolean

true = Yes

false = No

*user\_ack* = boolean

true = Yes

false = No

*address* = string

*text* = string (message body)

**Return Value:** handle or *invalid* or *error-code*

-100 = *text* parameter too long

-1 = unspecified error

**Associated Events:** A MessageSendStatus event occurs when the message has been sent.

**Exceptions:** If the *address* parameter is not a valid SNA, SSI or TSI as defined in 5.1.4, or a phone-number as defined in [GENFORMAT], this function returns *invalid*.

**Example:** `var handle = WTATETRA.sdsSend (4, 1, false, false, 5551234,"Example SDS");`

### 7.2.3 WTAITETRA.sdsSendAck

**Function:** `sdsSendAck (msgHandle, ack_type)`

**Function ID:** 2

**Description:** Sends an SDS Acknowledgement message in response to an incoming SDS message. See section 5.2 for description of the SDS Acknowledgement message.

The *msgHandle* parameter identifies the SDS message to be acknowledged.

This function returns an empty string if successful or *invalid* if it fails.

**Permission Types:** BLANKET, CONTEXT, SINGLE (see [WTA]).

**Parameters:**

`msgHandle` = handle

`Ack_type` = integer

0 = Delivery Ack

1 = User Ack

**Return value:** empty string (successful), *invalid* (failure indication) or `error-code` (-1 = unspecified error)

**Associated Events:** A MessageSent event occurs when the acknowledgement has been sent to the network.

**Exception:** If the *msgHandle* parameter does not refer to a network message for which information is available or, for which a User Acknowledgement was required, this function returns *invalid*.

If the Acknowledgement can not be sent e.g. because the outgoing message buffer is full, this function returns *invalid*.

**Example:** `var result = WTAITETRA.sdsSendAck (msgHandle);`

---

## Appendix A. WMLScript Library Functions

In the table below, WMLScript Function Libraries Calls valid for TETRA networks are summarised. The arguments have been left out in order to increase readability. The values in the column named "Lib/Func ID" denote the *Library* and *Function IDs*.

Lib/Func ID	WMLScript call	Description
522.0	WTATETRA.makeCall	Initiate a mobile originated, half or full duplex individual call, or half-duplex group call.
522.1	WTATETRA.sdsSend	Send a short data text message.
522.2	WTATETRA.sdsSendAck	Send acknowledgement to a short data text message.

## Appendix B. Static Conformance Requirements

This static conformance clause defines a minimum set of features that should be implemented to ensure that WTA could interact with the mobile network. A feature can be optional or mandatory. Although a function is mandatory it may not work, e.g. if the corresponding feature it is not implemented in the mobile or in the network or if the user has no subscription for this feature.

### B.1 Client Features

#### B1.1 TETRA Network Model

Item	Function	Reference	Status	Requirements
WTAITETRA-NMM-C-001	TETRA Voice Call Model	5.1	M	
WTAITETRA-NMM-C-002	TETRA Network Message Model	5.2	M	

#### B1.2 WTA Events

Item	WTA Event	Reference	Status	Requirements
WTAITETRA-E-C-001	Tx granted (wtaev-tetra/tg)	7.1.1	M	
WTAITETRA-E-C-002	Rx granted (wtaev-tetra/rg)	7.1.2	M	
WTAITETRA-E-C-003	Call queued (wtaev-tetra/cq)	7.1.3	M	
WTAITETRA-E-C-004	Released (wtaev-tetra/re)	7.1.4	M	
WTAITETRA-E-C-005	Incoming half duplex call (wtaev-tetra/ic)	7.1.5	M	
WTAITETRA-E-C-006	Received acknowledgement (wtaev-tetra/ra)	7.1.6	M	
WTAITETRA-E-C-007	Consumed acknowledgement (wtaev-tetra/ca)	7.1.7	M	
WTAITETRA-E-C-008	Call cleared (wtaev-cc/cl)	6.1.1	M	WTAI-CVE-C-002

#### B1.3 WMLScript Functions

Item	Function	Reference	Status	Requirements
WTAITETRA-C-001	WTATETRA.makeCall	7.2.1	M	
WTAITETRA-C-002	WTATETRA.sdsSend	7.2.2	M	WTAI-CTE-C-001
WTAITETRA-C-003	WTATETRA.sdsSendAck	7.2.3	M	WTAI-CTE-C-001

#### B1.4 WMLScript Bytecode Interpreter Capabilities

Item	Function	Reference	Status	Requirements
WTAITETRA-INT-C-001	Supports TETRA network WTAI library identifier	A	M	WMLS:MCF
WTAITETRA-INT-C-002	Supports TETRA network WTAI function identifier	A	M	WMLS:MCF

## B2. Server Features

### B2.1 WMLScript Encoder Capabilities

Item	Function	Reference	Status	Requirements
WTAITETRA-ENC-S-001	Supports TETRA network WTAI library identifier	A	M	WMLS:MSF
WTAITETRA-ENC-S-002	Supports TETRA network WTAI function identifier	A	M	WMLS:MSF



---

## Appendix C. Specification-track Document History

Type of Change	Date	Section	Description
Class 0	15-May-2001		The initial version of this document.