# Enabler Test Specification for Push
## Interoperability

Candidate Version 2.2 – 06 Nov 2007

**Open Mobile Alliance**

OMA-ETS-Push-V2_2-20071106-C

# Contents

# Figures

# 1. Scope

This document describes in detail the Interoperability test cases for Wap1 (WSP) and Wap2 (HTTP) OMA Push V2.2.The document is split into three sub sections covering the various Push 2.2 enablers:

- Push Access Protocol- (PAP)
- Push Proxy Gateway-(PPG)
- Over the Air - (OTA)

# 2. References

## 2.1 Normative References


**[ABNF]** "Augmented BNF for Syntax Specification: ABNF", D. Crocker, Ed., P. Overell. November 1997, URL: http://www.ietf.org/rfc/rfc2234.txt

**[DMSTDOBJ]** "OMA Device Management Standardized Objects, Version 1.2". Open Mobile Alliance™ OMA-TS-DM-StdObj-V1_2, URL:http://www.openmobilealliance.org

**[DM-TND-V1-2]** "OMA Device Management Tree and Description, Version 1.2". Open Mobile Alliance™ OMA-TS-DM_TND-V1_2 URL:http://www.openmobilealliance.org

**[HTML4]** "HTML 4.0 Specification, W3C Recommendation, revised on 24-Apr-1998", D. Raggett et al., April 24 1998. URL:http://www.w3.org/TR/1998/REC-html40-19980424

**[IOPProc]** "OMA Interoperability Policy and Process". Open Mobile Alliance™. OMA-IOP-Process-v1_5. URL:http//www.openmobilealliance.org/

**[ISO8601]** "Data elements and interchange formats - Information interchange - Representation of dates and times", International Organization For Standardization (ISO), 15-June-1988

"Data elements and interchange formats - Information interchange - Representation of dates and times, Technical Corrigendum 1", International Organization For Standardization (ISO) - Technical Committee ISO/TC 154, 01-May-1991

**[OMNA]** "OMA Naming Authority". Open Mobile Alliance™. URL:http://www.openmobilealliance.org/

**[PROVBOOT]** "Provisioning Bootstrap 1.1". Open Mobile Alliance™. OMA-WAP-ProvBoot-v1_1. URL:http://www.openmobilealliance.org

**[ProvCont]** "Provisioning Content Type Specification". Open Mobile Alliance™. WAP-183-ProvCont. URL: http://www.openmobilealliance.org/

**[PushMsg]** "Push Message Specification". Open Mobile Alliance™.. WAP-251-PushMessagea URL:http://www.openmobilealliance.org/

**[PushOTA V2.1]** "Push OTA Protocol", Open Mobile Alliance™. OMA-WAP-TS-PushOTA-V2_1 URL:http//www.openmobilealliance.org/

**[PushOTA V2.2]** "Push OTA Protocol", Open Mobile Alliance™. OMA-WAP-TS-PushOTA-V2_2 URL:http//www.openmobilealliance.org/

**[PushOTA]** "Push OTA Protocol", Open Mobile Alliance™. OMA-WAP-TS-PushOTA-V2_1 URL:http//www.openmobilealliance.org/

**[PushPAP V 2.1]** "Push Access Protocol Specification". Open Mobile Alliance™. OMA-WAP-TS-PAP-V2_1 URL:http://www.openmobilealliance.org/

**[RFC1738]** "Uniform Resource Locators (URL)", T. Berners-Lee, et al., December 1994. URL: http://www.ietf.org/rfc/rfc1738.txt

**[RFC1951]** "DEFLATE Compressed Data Format Specification version 1.3". P. Deutsch. May 1996. URL: http://www.ietf.org/rfc/rfc1951.txt

**[RFC2046]** "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", N. Freed, et al. November 1996, URL: http://www.ietf.org/rfc/rfc2046.txt

**[RFC2119]** "Key words for use in RFCs to Indicate Requirement Levels". S. Bradner. March 1997. URL:http://www.ietf.org/rfc/rfc2119.txt

**[RFC2234]** "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell. November 1997. URL:http://www.ietf.org/rfc/rfc2234.txt

**[RFC2234]** "Augmented BNF for Syntax Specifications: ABNF". D. Crocker, Ed., P. Overell.

November 1997. URL: http://www.ietf.org/rfc/rfc2234.txt

**[RFC2387]** "The MIME Multipart/Related Content-type", E. Levinson, August 1998, URL:http://www.ietf.org/rfc/rfc2387.txt

**[RFC2616]** "Hypertext Transfer Protocol – HTTP/1.1", R. Fielding, et al. June 1999, URL:http://www.ietf.org/rfc/rfc2616.txt

**[RFC3513]** "IP Version 6 Addressing Architecture". R. Hinden, et al. July 1998. URL: http://www.ietf.org/rfc/rfc3513.txt

**[RFC791]** "Internet Protocol". J. Postel. September 1981. URL: http://www.ietf.org/rfc/rfc791.txt

**[RFC822]** "Standard for the Format of ARPA Internet Text Messages". David H. Crocker. August 1982. URL: http://www.ietf.org/rfc/rfc822.txt

**[UAPROF]** "Wireless Application Group User Agent Profile Specification", Open Mobile Alliance™, WAP-248-UAProf. URL: URL: http://www.openmobilealliance.org/

**[WBXML]** "Binary XML Content Format Specification". Open Mobile Alliance™. WAP-192-WBXML URL:http://www.openmobilealliance.org/

**[WDP]** "Wireless Datagram Protocol". Open Mobile Alliance™. WAP-259-WDP. URL:http://www.openmobilealliance.org/

**[WSP]** "Wireless Session Protocol". Open Mobile Alliance™ WAP-230-WSP. URL: http://www.openmobilealliance.org/

**[XML]** "Extensible Markup Language (XML) 1.0 (Second Edition)", W3C Recommendation 6-October-2000. T. Bray, et al, 6-October-2000. URL: http://www.w3.org/TR/REC-xml

## 2.2 Informative References

**[ERELDDM]** "Enabler Release Definition for Device Management version 1.2". Open Mobile Alliance™, OMA-ERELD-DM-V1_2-20060208-C.

**[PROVARCH]** "Provisioning Architecture Overview 1.1". Open Mobile Alliance™.OMA-WAP-ProvArch-v1_1. URL:http://www.openmobilealliance.org/

**[PushArch]** "WAP Push Architectural Overview", Open Mobile Alliance™, WAP-250-PushArchOverview, URL:http://www.openmobilealliance.org/

**[PushMsg]** "Push Message Specification", Open Mobile Alliance™, WAP-251-PushMessage, URL: URL:http://www.openmobilealliance.org/

**[RDF]** "Resource Description Framework (RDF) Model and Syntax Specification", W3C Recommendation, 22-February-1999. URL: http://www.w3.org/TR/REC-rdf-syntax

**[RFC2246]** "The TLS Protocol Version 1.0", T. Dierks, C. Allen. January 1999, URL:http://www.ietf.org/rfc/rfc2246.txt

**[RFC2396]** "Uniform Resource identifiers (URI)", T. Berners-Lee, et al., August 1998. URL: http://www.ietf.org/rfc/rfc2396.txt

**[RFC822]** "Standard for the Format of ARPA Internet Text Messages", D. Crocker, August 1982. URL: http://www.ietf.org/rfc/rfc0822.txt

**[WAPARCH]** "WAP Architecture". Open Mobile Alliance™. WAP-210-WAPArch. URL: http://www.openmobilealliance.org/

# 3. Terminology and Conventions

## 3.1 Conventions

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].

All sections and appendixes, except “Scope”, are normative, unless they are explicitly indicated to be informative.

The following numbering scheme is used:

**xxx-y.z-con-number** where:

| | |
|---|---|
| xxx | Name of enabler, e.g. MMS or Browsing |
| y.z | Version of enabler release, e.g. 1.2 or 1.2.1 |
| ’con’ | Indicating this test is a conformance test case |
| number | Leap number for the test case |

Or

**xxx-y.z-int-number** where:

| | |
|---|---|
| xxx | Name of enabler, e.g. MMS or Browsing |
| y.z | Version of enabler release, e.g. 1.2 or 1.2.1 |
| ’int’ | Indicating this test is a interoperability test case |
| number | Leap number for the test case |

## 3.2 Definitions

| | |
|---|---|
| **Application** | A value-added data service provided to a Client. The application may utilise both push and pull data transfer to deliver content |
| **Application Addressing** | **T**he ability to address a particular user agent on a WAP client. |
| **Bearer Network** | a network used to carry the messages of a transport-layer protocol between physical devices. Multiple bearer networks may be used over the life of a single push session. |
| **Client** | In the context of push, a client is a device (or service) that expects to receive push content from a server. In the context of pull a client, it is a device initiates a request to a server for content or data. See also "device" |
| **Contact Point** | Address information that describes how to reach a push proxy gateway, including transport protocol address and port of the push proxy gateway. |
| **Content** | Subject matter (data) stored or generated at an origin server. Content is typically displayed or interpreted by a user agent on a client. Content can both be returned in response to a user request, or pushed directly to a client. |
| **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. |
| **Device** | Is a network entity that is capable of sending and/or 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,<br>a device can service a number of clients (as a server) while being a client to another server. |
| **End-user** | See "user" |
| **Multicast Message** | a push message containing a single address which implicitly specifies more than one OTA client address. |
| **Push Framework** | The entire Push system. The push framework encompasses the protocols, service interfaces, and |

| | |
|---|---|
| | software entities that provide the means to push data to user agents in the WAP client. |
| **Push Initiator** | The entity that originates push content and submits it to the push framework for delivery to a user agent on a client. |
| **Push OTA Protocol** | A protocol used for conveying content between a Push Proxy Gateway and a certain user agent on a client. |
| **Push Proxy Gateway** | A proxy gateway that provides push proxy services. |
| **Push Session** | A WSP session that is capable of conducting push operations. |
| **Registration** | Refers to a procedure where the PPG becomes aware of the terminal's current capabilities and preferences. |
| **Registration Context** | A state where the PPG is aware of at least the last capabilities and preferences conveyed from the terminal. |
| **Server** | A device (or service) that passively waits for connection requests from one or more clients. A server may accept or reject a connection request from a client. A server may initiate a connection to a client as part of a service (push). |
| **Terminal** | See "client". |
| **Terminal-ID** | An identifier that is used by a PPG to uniquely identify a terminal. |
| **User** | A user is a person who interacts with a user agent to view, hear, or otherwise use a rendered content. Also referred to as end-user |
| **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. |

## 3.3 Abbreviations

| | |
|---|---|
| **ABNF** | Augmented Backus-Naur Form |
| **ABNF** | Augmented Backus-Naur Form |
| **CPI** | Capability and Preference Information |
| **CSD** | Circuit Switched Data |
| **DNS** | Domain Name Server |
| **DTD** | Document Type Definition |
| **ETR** | Enabler Test Requirements |
| **ETS** | Enabler Test Specification |
| **GPRS** | General Packet Radio Service |
| **HTTP** | Hypertext Transfer Protocol |
| **IANA** | Internet Assigned Numbers Authority |
| **IP** | Internet Protocol |
| **MAC** | Authenication code |
| **MS** | Mobile Station |
| **MSISDN** | Mobile Station International Subscriber Directory Number |
| **OMA** | Open Mobile Alliance |
| **OMA** | Open Mobile Alliance |
| **OMNA** | Open Mobile Naming Authority |
| **OTA** | Over The Air |
| **OTA-HTTP** | (Push) OTA over HTTP |

| | |
|---|---|
| **OTA-HTTP-TLS** | OTA-HTTP over TLS |
| **OTA-WSP** | (Push) OTA over WSP |
| **PAP** | Push Access Protocol |
| **PDP** | Packet Data Protocol |
| **PI** | Push Initiator |
| **PO-TCP** | PPG Originated TCP connection establishment method |
| **PPG** | Push Proxy Gateway |
| **QoS** | Quality of Service |
| **RADIUS** | Remote Authentication Dial-In User Service |
| **RFC** | Request For Comments |
| **SEC** | Security Control |
| **SHA-1** | Secure Hash Algorithm 1 |
| **SI** | Service Indication |
| **SIA** | Session Initiation Application |
| **SIR** | Session Initiation Request |
| **SL** | Service Loading |
| **SMS** | Short Message Service |
| **TCP** | Transmission Control Protocol |
| **TLS** | Transport Layer Security |
| **TO-TCP** | Terminal Originated TCP connection establishment method |
| **UDP** | User Datagram Protocol |
| **URI** | Uniform Resource Identifier |
| **URL** | Uniform Resource Locator |
| **WAP** | Wireless Application Protocol |
| **WBXML** | WAP Binary XML |
| **WDP** | Wireless Datagram Protocol |
| **WINA** | WAP Interim Naming Authority |
| **WSP** | Wireless Session Protocol |
| **WTLS** | Wireless Transport Layer Security |

# 4. Introduction

The purpose of this document is to provide test cases for all the Push 2.2 Enabler.

Any issues found during Conformance testing SHOULD be reported to OMA using Enabler Test Reports.

The following items are needed to test Push 2.2

- A Push initiator Tool to connect to the PPG for initating the Push messsage. The Push initiator should be able to represent various types of application ID. Ie MMS, DRM, Device Provisioning etc…
- A Push Proxy Gateway (PPG) supporting WSP/HTTP Push, this facilitates push delivery from the wired to wireless network.
- A Protocol Analyzer to monitor and diagnose the connections between the systems.
- A Push capable client which can support all or any of the following (Connectionless/Connection Oriented via WSP/HTTP with secure or unsecure support)
- A provisioning server to pre provision security Push server settings if required.

The following Block Diagram helps to portray the overall system architecture:

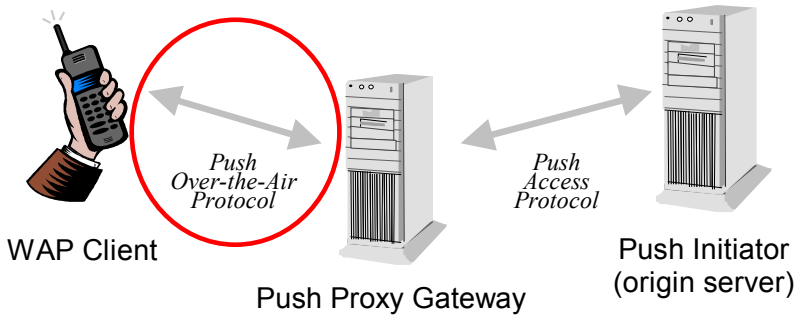
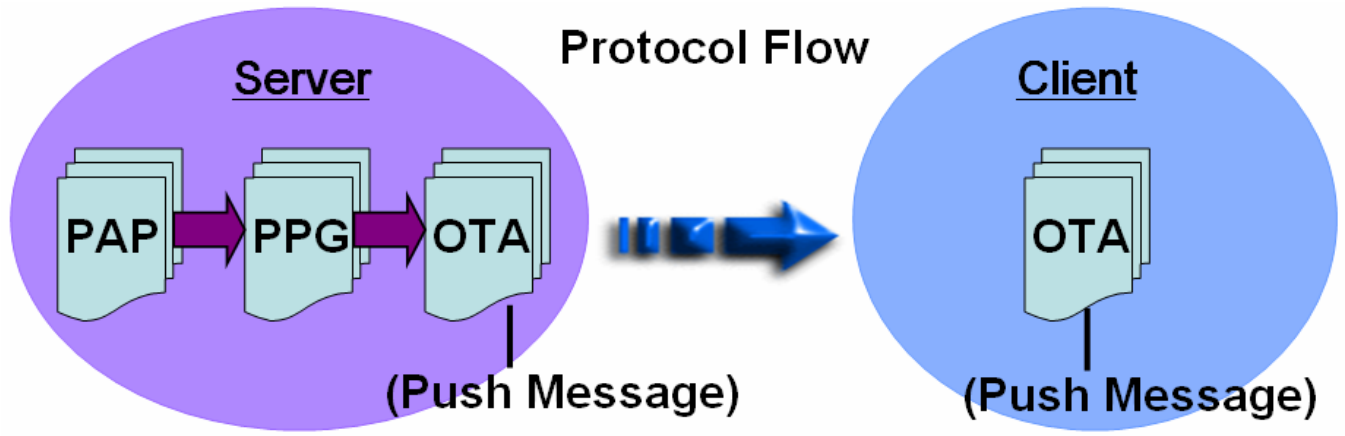



**Figure 1: System architecture**

To help understand the various subsections and protocols used Figure 2 below shows the protocol flow.

**Figure 2: System Protocol Flow**




## 4.1 Security Considerations

In order to protect against denial of service attacks and push from unauthorized sources a 'whitelist' mechanism is defined. If the whitelist is provisioned on the terminal, then it MUST be used to validate the source address of Push PDUs received over connectionless bearers. However if the whitelist is not provisioned then the mobile device MUST accept the push PDU and process it according to its content.

## 4.2 Processing Session Initiation Requests

In the case of SIR, to protect against denial of service attacks, the terminal SHOULD implement a lockout timer. If the terminal receives any additional SIRs during the lockout interval, it should defer processing or discard them until the timer expires. If the requested push session(s) is successfully established (OTA-WSP), or if the active TCP connection(s) is successfully established (OTA-HTTP), the lockout timer SHOULD be reset. The value of the lockout timer interval is implementation specific.

To protect against spoofing, the terminal SHOULD validate the SIR by comparing the source address of the PDU that carries the SIA content with a pre-existing list of authorised PPGs. The SIR SHOULD be ignored if the validation fails.

The above measures are applicable if the SIR is received on a non-secure port. If a secure port is used, these measures are generally not necessary.

## 4.3 Processing content sources in a connectionless environment

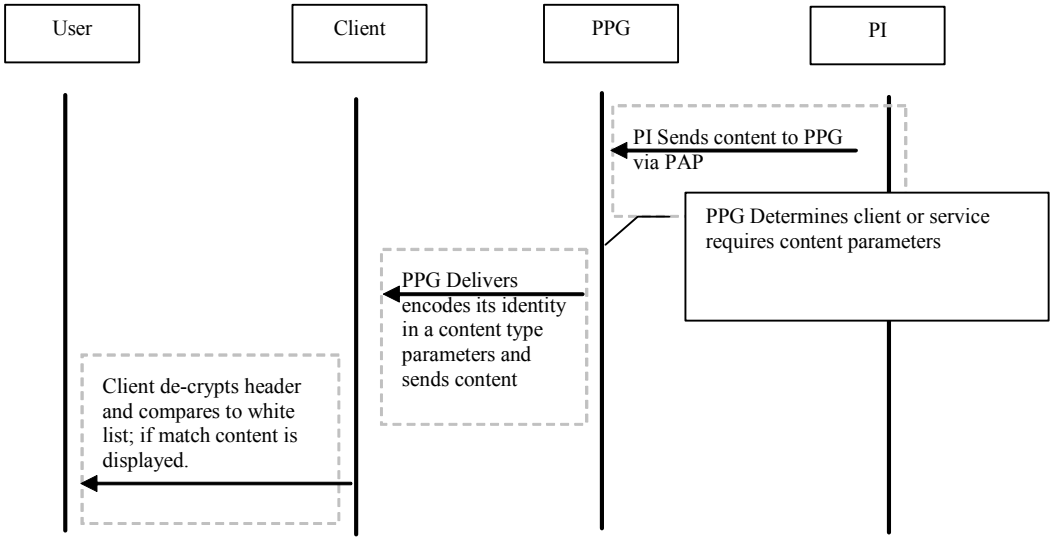
A secondary mechanism MAY be used to verify the originating source of the content (either PI or PPG). In order to establish & authenticate a trust relationship between PPG and client the mechanism detailed in the Provisioning Bootstrap [ProvBoot] is reused. In Bootstrap the content is trusted due to it being 'signed' using a

shared secret. This shared secret may be user defined or it might be some specific information that is related to the bearer or network. The shared secret is then used to generate parameters to the provision content type *application/vnd.wap.connectivity-wbxml*; namely the parameters SEC and MAC.

SEC indicates the security mechanism that was used (user defined, network specific etc) and the MAC parameter indicates the authentication code calculated using the pushed data and the shared secret.

In the case where the shared secret is known to the PPG (or PI) the PPG (or PI) may generate the SEC and MAC parameters for the content type.

The use of SEC and MAC is as per OMA Provisioning Bootstrap specification [ProvBoot]. The following example clarifies the usage in the case of Push OTA




SEC value

The SEC parameter can have the following values

| Value | Meaning |
|---|---|
| 1 | USERPIN |
| 2 | USERNETWPIN |
| 3 | USERPINMAC |

MAC value

Defined as follows: -

MAC = HMAC-SHA (K, D)

Where K is the Shared Secret and D is the data, in this case the content that is to be pushed to the targeted terminal and HMAC-SHA is the Keyed-Hashing for Message Authentication algorithm [RFC 2104], which utilizes the SHA-1 hash function.

If the terminal does not understand the added content-type parameters, it MUST ignore them. However, if the terminal is capable of processing the parameters to the content type and they do not match the shared SEC parameter value the push MUST be rejected.

# 5. Conformance Test Cases

Not available

# 6. OTA Test Cases

There are 40 interoperability test cases for Push 2.2 OTA Push Enabler.

## 6.1 WSP Server/Client Push OTA

### 6.1.1 Non-Secure Port for Connectionless Push

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-1 |
| **Test Object** | Server/client device |
| **Test Case Test Case Description and Purpose** | Verify that a non-secure connectionless SI/SL Push message is sent by the server and received by the client. |
| **Specification Reference** | [PushOTA] Section 5<br>[PushOTA] Section 6.2.1 |
| **SCR Reference** | WSP-CL-C-002 AND<br>WSP-CL-C-003 AND<br>WSP-CL-C-020 AND<br>WDP-RP-S-004<br>OTA-SEC-C-001 AND<br>WDP:MCF |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile with security set to “OFF”.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test procedure** | Set the right WAP Profile (with PPG IP address) on your Client and select it as the current one.<br>Send the push message to the Client. The bearer delivering the push message can be either SMS or an unsecure WSP Session. When the Client displays reception of the message, the user will then select to download the push message.<br><br>When the Device connects to the PPG the Message that was waiting for the device is retrieved. |
| **Pass -Criteria** | The Client can receive the push message via SMS or via a Non Secure WAP session. Where upon the received PUSH is in the inbox and the end-user can load it successfully. |
| **Comment** | This test case should be executed both when the Client is in idle mode and when it has a Non Secure WAP session using both the Service Indication and Service Loading. |

## 6.1.2 Connection-Oriented Unconfirmed Push

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-2 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | To verify that a connection oriented unconfirmed SL/SI Push message is sent by the server and received by the client.<br>If the client has not established a bearer then the SIR is sent via OTA until the device connects via the bearer and the session is active<br><br>A SIR will be sent first followed by a Push SI/SL message. |
| **Specification Reference** | [PushOTA] Section 6.2.2<br>[PushOTA] Section 5<br>[PushOTA] Section 8.3 |
| **SCR Reference** | OTA-WSP-C-002 (WSP-CO-C-001 and WSP-CO-C-010), OTA-WSP-C-003<br>OTA-CO-C-002 (OTA-WSP-C-001 or and OTA-WSP-C-002) and (OTA-WSP-C-003 or OTA-WSP-C-004: wtls:mcf and wtls:wtls-c007) and OTA-WSP-C-005 and OTA-CO-001 (OTA-CO-C-002 or OTA-CO-C-003)<br>OTA-WSP-C-011<br>OTA-WSP-S-002 (WSP-CO-S-001 and WSP-CO-S-010)<br>OTA-CO-S-002<br>OTA-SEC-C-001 AND OTA-HTTP-S-001 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PPG Server will send an SI and SL Push messages to the client via the active bearer “Wap session”. If the client has not established the bearer the SIR is sent via OTA until the device connects via the bearer and the session is active, then the message content is delivered over the session. |
| **Pass-Criteria** | The Client initiates a WAP session, receives the push message and ends the WAP session successfully. The received PUSH SL/SI can be presented successfully.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing WAP connection. Should also be executed using both SI and SL messages and taking into consideration security issues. |

## 6.1.3 Connection-Oriented Confirmed Push

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-3 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify Connection Oriented Confirmed Push by sending a confirmed Push request from the PI which sub sequentially the PPG will then send a connection oriented confirmed SI/SL Push message and this is successfully received by the client when it is on idle mode.<br>If the client has not established a bearer then the SIR is sent via OTA until the device connects via the bearer and the session is active<br><br>A SIR will be sent first followed by a Push SI/SL message |
| **Specification Reference** | [PushOTA] Section 6.2.2<br>[PushOTA] Section 5 |
| **SCR Reference** | (OTA-WSP-C-001 OR<br>OTA-WSP-C-002) AND<br>(OTA-WSP-C-003 OR<br>OTA-WSP-C-004) AND<br>OTA-WSP-C-005 AND<br>OTA-SEC-C-001 AND OTA-HTTP-S-001<br>WSP-CO-S-001 AND WSP-CO-S-011 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PI will send a Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via the active Wap session. If the client has not established the session the message is then queued until the device connects via the session again. The PPG server may send another SIR via OTA to prompt the phone to reconnect. |
| **Pass-Criteria** | The Client initiates a WAP session, receives the push message and ends the WAP session successfully. The received PUSH SL/SI can be presented successfully.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing WAP connection.<br>Should also be executed using both SI and SL messages |

### 6.1.4 Support for 4 Concatenated SMS's

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-4 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client can support concatenating 4 segmented SMS messages. |
| **Specification Reference** | [PushOTA] Section 6.2<br>[PushOTA] Section 6.2.1.1 |
| **SCR Reference** | WDP-CDMA_C-001 OR<br>WDP-GSM-C-001<br>WDP-CDMA-S-001 OR<br>WDP-GSM-S-001 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via 4 segmented SMS's to transmit the payload. The Client will then process the Push message by concatenating the push messages. |
| **Pass-Criteria** | The client will be concatenating the messages to formulate the completed message payload. The client will also validate the push message source address against the Whitelist. If successful the push message will be processed otherwise it will be ignored. If no whitelist is defined then by default the Push will be accepted. The full Payload will then be displayed successfully. |
| **Comments** | |

### 6.1.5 Support for Whitelists

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-5 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client configured with a whitelist mechanism must validate the source address of the Push PDU's received over the connectionless bearer. If no whitelist is configured then the device will accept any push. |
| **Specification Reference** | [PushOTA] Section 8.3<br>[PushOTA] Section 8.3.1 |

| | |
|---|---|
| **SCR Reference** | OTA-SEC-C-001<br>OTA-SEC-C-003 OR<br>OTA-SEC-C-004 OR |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The Client will then process the Push messge. |
| **Pass-Criteria** | The client will validate the push message source address against the Whitelist. If successful the push message will be processed otherwise it will be ignored. If no whitelist is defined then by default the Push will be accepted. |
| **Comments** | |

### 6.1.6 Secondary Source Authenication Connectionless

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-6 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client configured with a trust mechanism must validate the originating source of content of the Push PDU's received over the connectionless bearer. If no trust is defined then the device will accept any push. |
| **Specification Reference** | [PushOTA] Section 8.3.3<br>[PushOTA] Section 8.3.3.2 |
| **SCR Reference** | OTA-SEC-C-002<br>OTA-SEC-C-007 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The Client will then process the Push messge. |

| | |
|---|---|
| **Pass-Criteria** | The client will validate the push message source address against the defined SEC & MAC. If successful the push message will be processed otherwise it will be ignored. If no SEC & MAC are defined then by default the Push will be accepted. |
| **Comments** | |

### 6.1.7 Support for Whitelists via Wap Provisioning

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-7 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client can be provisioned with a Whitelist via the OMA Provisioning protocol [Prov Cont] using the VENDORCONFIG parameters<br>If no Provisioning is applied then no trust is defined so then the device will accept any push. |
| **Specification Reference** | [PushOTA] Section 8.3.1 |
| **SCR Reference** | OTA-SEC-C-003<br>[Prov – Cont] |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs.<br>The Provisoning Whitelist Vendor Config parameters are sent to the device. Where up they are used to validate the Push PDU.<br><br>In the case of VENDORCONFIG each parameter in the push whitelist will be named `WHITE_LISTn_SME` and `WHITE_LISTn_SMSC` for matched values of *n*, which runs from 1 through the maximum allowed number of entries. Each value in the whitelist MUST support specification of both an SME (Short Message Entity -- source number) and an SMSC (Short Message Service Center) through which the SME sends messages. Each character in an SME or SMSC address in a whitelist entry MUST be in the set {0..9, #, *, A, B, C, a, b, c} |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The Client will then process the Push messge. |
| **Pass-Criteria** | On receipt of a Push PDU the client SHOULD verify the origination address of the PDU against the push whitelist. When matching the address against a whitelist entry, the client MUST support prefix matching, i.e. with the address formed into a string of the form <number-plan><type-of-number><digit>+, the address matches the corresponding half of the whitelist entry if all the characters in the entry match all the leading characters of the address. |

| | |
|---|---|
| **Comments** | |

## 6.1.8 Support for Whitelists via Device management Object

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-8 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client can be provisioned with a Whitelist via the OMA Provisioning protocol [ERELDDM] using an extension to the DM Tree {DM-TND-V1.2] as a management object DMSTOBJ<br>If no Provisioning is applied then no trust is defined so then the device will accept any push. |
| **Specification Reference** | [PushOTA] Section 8.3.1 |
| **SCR Reference** | OTA-SEC-C-004<br>[DMSTDOBJ] |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs.<br>The Provisoning Whitelist Vendor Config parameters are sent to the device. Where up they are used to validate the Push PDU.<br><br>In the case of VENDORCONFIG each parameter in the push whitelist will be named `WHITE_LIST`*n*`_SME` and `WHITE_LIST`*n*`_SMSC` for matched values of *n*, which runs from 1 through the maximum allowed number of entries. Each value in the whitelist MUST support specification of both an SME (Short Message Entity -- source number) and an SMSC (Short Message Service Center) through which the SME sends messages. Each character in an SME or SMSC address in a whitelist entry MUST be in the set {0..9, #, *, A, B, C, a, b, c} |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The Client will then process the Push messge. |
| **Pass-Criteria** | On receipt of a Push PDU the client SHOULD verify the origination address of the PDU against the push whitelist. When matching the address against a whitelist entry, the client MUST support prefix matching, i.e. with the address formed into a string of the form <number-plan><type-of-number><digit>+, the address matches the corresponding half of the whitelist entry if all the characters in the entry match all the leading characters of the address. |
| **Comments** | |

### 6.1.9 Support for Whitelists via Wap Provisioning (PXADDR)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-9 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client can be provisioned with a Whitelist via the Physical Proxies settings provisioned on the device. |
| **Specification Reference** | [PushOTA] Section 8.3.1 |
| **SCR Reference** | OTA-SEC-C-005<br>[Prov – Cont] |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | If no entry matches in the configured whitelists the terminal MAY compare the origination address data to the push enabled physical proxies provisioned on the device (PXPHYSICAL [ProvCont]), for example If the push mechanism is SMS based the source address may be the SMSC Number., this may be compared to a PXADDR of type E164. In an alternate example the PXADDR could be of type IPv4 for comparison with the source IP connectionless push PDUs transported over UDP. In the event of this being a WAP 2 Push where the Push Proxy gives guidance to the device as to which provisioning context to use, via the X-Wap-Push-ProvURL then that context MUST be used.<br><br>If origination address does not match any address configured in the whitelist or other connectivity configuration the terminal MUST reject the push PDU. |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The Client will then process the Push messge. |
| **Pass-Criteria** | On receipt of a Push PDU the client SHOULD verify the origination address of the PDU against the push whitelist. If it matches then the Push message is accepted otherwise it is rejected. If no whitelist configuration is set then the message by default will be accepted |
| **Comments** | |

### 6.1.10 Lockout Timer Support for SIA

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-10 |
| **Test Object** | Server/client device |

| | |
|---|---|
| **Test Case Description and Purpose** | Verify that the Client can support a lockout timer to protect against denial of service attacks. |
| **Specification Reference** | [PushOTA] Section 8.3.2 |
| **SCR Reference** | OTA-SEC-C-006 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | In the case of SIR, to protect against denial of service attacks, the terminal SHOULD implement a lockout timer. If the terminal receives any additional SIRs during the lockout interval, it should defer processing or discard them until the timer expires. If the requested push session(s) is successfully established (OTA-WSP), or if the active TCP connection(s) is successfully established (OTA-HTTP), the lockout timer SHOULD be reset. The value of the lockout timer interval is implementation specific. |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via TO TCP. The Client will then process the initial SIR Push message. When accepted for processing by the device, attempt to resend a few more to ensure that they are rejected by the device during the Lock out period.<br>The timer will reset upon successful connection. Then resend another SIR which will be accepted by the device. |
| **Pass-Criteria** | Ensure the device accepts the first SIR and rejects subsequent SIRs. Until the connection has been established where upon the Lockout timer resets and and the client will accept another SIR again for the process to repeat itself. |
| **Comments** | |

## 6.1.11 Support for SEC/MAC Content Parameters

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-11 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | In order to establish & authenticate a trust relationship between PPG and client the mechanism detailed in the Provisioning Bootstrap [ProvBoot] is reused. In Bootstrap the content is trusted due to it being 'signed' using a shared secret. This shared secret may be user defined or it might be some specific information that is related to the bearer or network. The shared secret is then used to generate parameters to the provision content type *application/vnd.wap.connectivity-wbxml*; namely the parameters SEC and MAC.<br><br>SEC indicates the security mechanism that was used (user defined, network specific etc) and the MAC parameter indicates the authentication code calculated using the pushed data and the shared secret.<br><br>In the case where the shared secret is known to the PPG (or PI) the PPG (or PI) may generate the SEC and MAC parameters for the content type |

| | |
|---|---|
| | The SEC and the MAC parameters must have the same value as the values provisioned in the Device. |
| **Specification Reference** | [PushOTA] Section 8.3.3 |
| **SCR Reference** | OTA-SEC-C-007 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | |
| **Test Procedure** | The PI will send an Un Confirmed Push PAP message to the PPG<br>The PPG Server will then send an SI and SL Push messages to the client via SMS.<br>The SEC & MAC parameters are then calculated and will be added as parameters to the content type header value.<br>The content is then sent to the targeted terminal unencrypted with the new parameters added to the content type. |
| **Pass-Criteria** | The Client receives the Push message and will try and decode the message. If the client does not understand the added content-type parameters, it MUST ignore them. However, if the client is capable of processing the parameters to the content type and they do not match the shared SEC parameter value that is provisioned in the client.<br><br>Then the push MUST be rejected otherwise the push message is accepted. |
| **Comments** | |

### 6.1.12 OTA-WSP Session Initiated Request (SIR) and Session Initiated Application (SIA)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-12 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | To verify that the PPG server sends an Session Initiated Request (SIR) to the Client and the SIA in the client will service the SIR request. |
| **Specification Reference** | [PushOTA] Section 8, [PushOTA] Section 8.2, [PushOTA] Section 8.4, [PushOTA] Section 6.2.2 |
| **SCR Reference** | OTA-WSP-C-005, OTA-CO-C-002, OTA-WSP-S-005, OTA-CO-S-002 OTA-SEC-C-006 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The device<br>Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs.<br><br>If the secure session WTLS is requested by using the SIR secure port or a provisioned port then the client must ensure that a WTLS session exists before it creates a new push session. |
| **Test Procedure** | The PPG Server will send an SIR Push message to the client.<br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR, then the client should establish a push session towards one of the contact points in this case OTA WSP. It is left to the device to decide which protocol variant to use. However the SIR may indicate that it accepts any Application ID. Therefore the client has the responsibility to clean up the stale push sessions. |
| **Pass-Criteria** | The Client must accept the SIR and process the message by the SIA and the application ID. The client will carry out the following<br><br>The client must establish a connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-WSP defined in the SIR.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing WAP connection. Should also be executed using any application Push messages |

### 6.1.13 Application Addressing

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-13 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the server sends the correct X-WAP-Application_ID to service the supported client application.<br>Verify that the Client handles a push message which contains an X-Wap-Application-ID header set to a proper value and formatted in a proper way. |
| **Specification Reference** | [PushOTA] Section 6.2.3. |
| **SCR Reference** | OTA-WSP-C-006 and OTA-WSP-C-007<br>OTA-WSP-S-006 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set it in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | Send Connection less SI, SL and SIA to the Client |
| **Pass-Criteria** | The Client must accept and process the message by the appropriate application I.e. MMS, WapPush, DRM. The received Application PUSH can be presented successfully.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | The procedure should be executed for connectionless, connection oriented Push Service Indication, Service Loading and Session Initiation Application messages when the Client is on standby as well as during an ongoing WAP connection.<br><br>This test case should be executed under all conditions of the Push access user settings. This test case should also be executed using many different types of application ID Push messages. |

### 6.1.14 Application Dispatching

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-14 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client handles a push message by dispatching the received message to the appropriate Device application. |
| **Specification Reference** | [PushOTA] Section 6.2.3. |
| **SCR Reference** | OTA-WSP-C-006 and OTA-WSP-C-007<br>OTA-WSP-S-006 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set it in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | Send Connection less SI, SL and SIA to the Client |
| **Pass-Criteria** | The Client must accept and process the received message on the registered WDP port for connectless and for connection oriented over the session by dispatching the relevant message content to the appropriate application I.e. MMS, SMS, WapPush, and DRM.<br><br>The received Application PUSH must be presented successfully.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | The procedure should be executed for connectionless, connection oriented Push Service Indication, Service Loading and Session Initiation Application messages when the Client is on standby as well as during an ongoing WAP connection.<br><br>This test case should be executed under all conditions of the Push access user settings. This test case should also be executed using many different types of application ID Push messages. |

## 6.1.15 Push Initiator authentication using Authentication Flag

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-15 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Client handles in a proper way a push message which contains *Authenticated* flag clear and set.<br>Verify that the server sends a push message that contains authenticated flag and an Initiator URI. |
| **Specification Reference** | [PushOTA] Section 6.2.4. |
| **SCR Reference** | OTA-WSP-C-008<br>OTA-WSP-S-007 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The correct PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs |
| **Test Procedure** | The PPG Server will send an SI and SL Push messages to the client. The client will check the URI provided by the PPG Server with the one prestored on the device when the Authenticator Flag is set to True and the two URI matched then a trust has been established |
| **Pass-Criteria:** | The Client receives the WapPush but will check the following<br><br>When the Auth Flag set (TRUE) then URL is checked with prestored Push URL on device. If the check result is (TRUE) then the Push is loaded successfully.<br><br>When the Auth Flag set (TRUE) then URL is checked with prestored Push URL on device. IF the check result is (FALSE) then the Push is rejected.<br><br>When the Auth Flag set (FALSE) then URL is NOT checked and the Push is loaded successfully.<br><br>In the UDP logs verify that the push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing WAP connection.<br>Should also be executed using both SI and SL messages |

## 6.1.16 Bearer Selection and Control

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-16 |
| **Test Object** | Server/Client device |
| **Test Case Description and Purpose** | To verify that the PPG Server can read Bearer Selection as made by the Client when it connects to the Push Proxy Gateway. Also that the PPG Server can send the Last flag to the client for end of messages.<br><br>Verify that the Client handles the Bearer Selection and Control functionality in a proper way:<br>Bearer Selection: The Client sends the *Bearer-Indication* header when it initiates a WAP session.<br>Bearer Control: The Client ends the created session (initiated by a SIR message sent by the PPG) according to the *Last* flag status on the *Push-Flag* header included on the push message |
| **Specification Reference** | [PushOTA] Section 6.2.6 |
| **SCR Reference** | OTA-WSP-C-009 and OTA-WSP-C-010, OTA-WSP-S-009, OTA-WSP-S-008 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PPG Server will send an SI and SL Push messages to the client. The client when it connects will send a Bearer Type Indication so allowing the Server to select and use that appropriate bearer on the next push message. If this message is the last message then the Last Flag is set and the client knows no more messages are waiting. |
| **Pass-Criteria** | The Client receives the WapPush but with check the following<br><br>The Bearer Selection Flag will be defined when the client retrieves the message for future PPG reference.<br><br>The Last Flag is set by the Server if there are no more messages to send to the client. Therefore the client can tear down the session as it has received its end of messages.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing WAP connection. |

| | |
|---|---|
| | Should also be executed using both SI and SL messages. |

# 6.2 WSP Error Conditions

## 6.2.1 Application Addressing with an incorrect or non existing Application ID

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-17 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Application addressing to an incorrect application id<br><br>Verify that the Server can send and the Client handles a push message which contains an X-Wap-Application-ID header set to an incorrect value together with the appropriate content type in the proper format<br><br>Verify that the Client discards a push message that contains an X-Wap-Application-ID header set to a non existing application. Also verify that the client can handle this error condition. |
| **Specification Reference** | [PushOTA] Section 6.2.3 |
| **SCR Reference** | OTA-WSP-C-006 and OTA-WSP-C-007<br>OTA-WSP-S-006 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>Send a Service Initiation Application file in textual or numeric form<br>The right PPG IP address is set it in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an Errorous Service Initiation Application or none at all via Connection less SI, SL and SIA are sent to the MS. |
| **Pass-Criteria** | When the Service Indication file or Session Initiation Application file is sent to the Client, the Client should not react in any way. The push message is processed i.e. the Client should discard the Sir's because of their malformed, incorrect application ID If the Client receives no application Id then the Client can use the default one if it so chooses |
| **Comment** | The procedure should be executed for connectionless, connection oriented Push Service Indication, Service Loading and Session Initiation Application messages when the Client is on standby as well as during an ongoing WAP connection. |

## 6.2.2 SI with missing value attributes

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-18 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Management of messages without attributes.<br>Verify that the Client handles a received SI messages with some missing fields in a proper manner. |
| **Specification Reference** | [PushSI] Section 7<br>[PushSI] Section 6.2, 6.3<br>[PushSI] Section 7<br>[PushSL] Section 8<br>[PushSL] Section 6.1 |
| **SCR Reference** | SI-CF-C-001<br>SI-SEM-C-011<br>SI-CF-C-003<br>SI-VAL-S-001<br>SI-VAL-S-002<br>SI-VAL-S-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an SI with no CREATED and SI-EXPIRES dates to the Client.<br><br>Send an SI with no text for display to the Client<br><br>Send an SI with no SI-ID attribute value to the Client<br><br>Send an SI with no HREF attribute value to the Client<br><br>Send an SI with no ACTION attribute value to the Client |
| **Pass-Criteria** | The Client should be able to process an SI with no CREATE and SI-EXPIRES attribute value i.e. the message is stored in the inbox and can be presented.<br><br>The Client should be able to process an SI with no test for display i.e. the message will be stored in the push inbox and no text is displayed.<br><br>The Client should be able to process an SI with no SI-ID attribute value i.e. the message is stored in the Push inbox and can be presented.<br><br>The Client should be able to process an SI with no HREF attribute value i.e. the message is stored in the Push inbox but it will not be possible to load the message. A consistent error message should be displayed (e.g. wrong URL).<br><br>The Client should be able to process an SI with no ACTION value i.e.the.SI shall be presented with for example the following text: “Push message received”, “No |

| | |
|---|---|
| | ACTION” |
| **Comment** | None. |

## 6.2.3 SL with missing value attributes

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-19 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Management of messages without attributes.<br>Verify that the Client handles a received SL messages with some missing fields in a proper manner. |
| **Specification Reference** | [PushSI] Section 7<br>[PushSI] Section 6.2, 6.3<br>[PushSI] Section 7<br>[PushSL] Section 8<br>[PushSL] Section 6.1 |
| **SCR Reference** | SL-CF-C-001 and SL-CF-C-002<br>SL-SEM-C-001 and SL-SEM-C-002<br>SL-VAL-S-001<br>SL-VAL-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an SL with no ACTION attribute value to the Client<br><br>Send an SL with no HREF attribute value to the Client. The Push initiator Tool may not allow the creation of an SL without a HREF. |
| **Pass-Criteria** | The Client should be able to process an SL message without ACTION value. The default values supported by the Client will be used.<br><br>The client should NOT be able to process an SL message with no HREF value, there will be no indication of reception of the message and therefore no message will be stored in the Push inbox. |
| **Comment** | None. |

# 6.3 WSP Server/Client Cache Operation

## 6.3.1 Support for the CO in tokenized form using URI Equivalence Rules and Prefix Match Rules

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-20 |
| **Test Object:** | Client and server devices |
| **Test Case Description and Purpose** | To verify that content is sent from the server and loaded to cache with a Push SL/SI. When the Client receives a valid cache operation message, the content is removed from the cache.<br>Verify that when the Client receives a valid invalidate-object push message it will delete the object following the URI Prefix Match rules, which include:<br>• The scheme must be the same.<br>• The authority must be the same.<br>• The path must match.<br>• The query part is ignored if included.<br>And the URI Equivalence rules include the following:<br>• A comparison of host name must be case insensitive.<br>• A port that is empty or not given is equivalent to the default port for that URI-reference.<br>• Comparison of scheme names must be case-insensitive.<br>• An empty absolute path is equivalent to an abs_path of "/". |
| **Specification Reference** | [PushCO] Section 9<br>[PushCO] Section 9.3<br>[PushCO] Section 6<br>[PushCO] Section 6.3<br>[PushCO] Section 6.4 |
| **SCR Reference** | CO-CF-C-001<br>CO-CF-C-002<br>CO-CF-C-003<br>CO-SEM-C-001<br>CO-SEM-C-003<br>CO-SEM-C-004<br>CO-PPG-S-001<br>CO-PPG-S-002<br>CO-PPG-S-003<br>CO-VAL-S-001<br>CO-VAL-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either "Always ask", "Always" or "Never".<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an SL message is sent to the Client. |
| **Pass-Criteria** | Send a Cache Operation message which will invalidate the cache for that URL |

| | |
|---|---|
| | previously sent.<br><br>Then send another SL with a different URL.<br><br>If the last SL results in the content being retrieved via the Wap pull channel as detectable from the protocol analyzer, then the test passes |
| **Comment** | This is tested both in idle mode and in non-idle mode using SL. |

## 6.3.2 URI Resolution in the invalidate object and service

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-21 |
| **Test Object** | Client Device |
| **Test Case Description and Purpose** | Resolution of relative URI when receiving invalidate-object or service.<br>Verify that when the Client receives an invalidate-object or invalidate-service message it only deletes those messages following the URI Resolution rules, which includes:<br>• If X-Wap-Content-URI exists in document it uses this to resolve the URI.<br>• If content-Location exists in document it uses this to resolve the URL.<br>• If "host" exists the Client does not need to use "host" header to resolve the URL.<br>• If not possible to resolve the URL, the CO is silently discarded. |
| **Specification Reference** | [PushCO] Section 6<br>[PushCO] Section 6.2<br>[PushCO] Section 6.3<br>[PushCO] Section 6.4 |
| **SCR Reference** | CO-SEM-C-002<br>CO-SEM-C-003<br>CO-SEM-C-004<br>CO-SEM-C-005 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either "Always ask", "Always" or "Never".<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | An SL message is sent to the Client. |
| **Pass-Criteria** | A page with a Cache Operation message with an invalidate object or service is loaded and the messages will be processed according to the URI resolution rules below:<br><br>If X-Wap-Content-URI in push message was not the correct one, therefore the object or service is not downloaded but it is taken from the cache.<br><br>Content location should not be used to resolve an invalidate service or object, i.e. the object or service is loaded from the cache<br><br>If host header is used, the push message will be discarded silently since host header is not used to resolve the URL.<br><br>If the absolute path is empty, then nothing will happen. |
| **Comment** | This is tested both in idle mode and in non-idle mode using both SI and SL. |

## 6.3.3 Protection for the denial of Service attacks

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-22 |
| **Test Object** | Client device |
| **Test Case Description and Purpose** | Verify that when the Client receives an invalidate-object or invalidate-service message it only deletes those messages following the URI Resolution rules, which includes:<br>• If X-Wap-Content-URI exists in document it uses this to resolve the URI.<br>• If content-Location exists in document it uses this to resolve the URL.<br>• If "host" exists the Client does not need to use "host" header to resolve the URL.<br>• If not possible to resolve the URL, the CO is silently discarded. |
| **Specification Reference** | [PushCO] Section 6.2<br>[PushCO] Section 6 |
| **SCR Reference** | CO-SEM-C-005<br>CO-SEM-C-001 and CO-SEM-C-002<br>OTA-SEC-C-006 AND<br>OTA-SEC-C-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Prerequisites:** | Push inbox and cache content are empty.<br>Push access user setting is set to either "Always ask", "Always" or "Never".<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test procedure:** | An SL message is sent to the Client. |
| **Pass criteria:** | A page with a Cache Operation message with an invalidate object or service is loaded and processed according to the rules below:<br><br>If X-Wap-Content-URI in push message was not the correct one, therefore the object or service is not downloaded but it is taken from the cache.<br><br>Content location should not be used to resolve an invalidate service or object, i.e. the object or service is loaded from the cache<br><br>If host header is used, the push message will be discarded silently since host header is not used to resolve URL.<br><br>If the absolute path is empty, then nothing will happen. |
| **Comment** | This is tested both in idle mode and in non-idle mode using both SI and SL. |

# 6.4 Service Indication

## 6.4.1 Character encoding

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-23 |
| **Test Object** | Client/Server devices |
| **Test Case Description and Purpose** | Verify that the sent and received SI messages with the supported encodings are processed by the Client in the correct way. |
| **Specification Reference** | [PushSI] Section 5.1<br>[PushSI] Section 7<br>[PushSI] Section 8 |
| **Requirement:** | SI-CSE-C-005<br>SI-CSE-C-006.<br>SI-CF-C-001<br>SI-CF-C-004<br>SI-PPG-S-001<br>SI-PPG-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either “Always ask”, “Always” or “Never”.<br>Current date / time on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send some SIs with the supported encoded characters to the Client. |
| **Pass-Criteria** | The Client should be able to load the SI message |
| **Comment** | None |

## 6.4.2 Support for %Datetime; encoded as OPAQUE data

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-24 |
| **Test Object** | Client device |
| **Test Description and Purpose** | Verify that the MS uses the SI-EXPIRES attribute to remove expired Service Indication messages from the Inbox and silently ignore the messages that have a Service Indication-EXPIRES date/time which is less than the date/time set on the Client. |
| **Specification Reference** | [PushSI] Section 8.2.1.1<br>[PushSI] Section 8.2.2<br>[PushSI] Section 6.2, 6.5<br>[PushSI] Section 7<br>[PushSI] Section 8<br>[PushSI] Section 6.4 |
| **SCR Reference** | SI-CF-C-005<br>SI-CF-C006<br>SI-SEM-C-002<br>SI-CF-C-001<br>SI-CF-C004<br>SI-SEM-C-015, SI-SEM-C-016 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send the Service Indication message to the Client with an SI-EXPIRES date/time more than the current one set on the Client.<br>Send a Service Indication message to the Client with an SI-EXPIRES date/time less than the current one set on the Client. |
| **Pass-Criteria** | SI messages with specified SI-EXPIRES attribute should be deleted from the inbox when the time expires and messages send when the SI-EXPIRES attribute has already passed should be ignored by the Client i.e. should not be stored in the Push inbox. |
| **Comment** | None. |

## 6.4.3 Handling of Out of Order SI and Replacement

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-25 |
| **Test Object** | Client device |
| **Test Case Description and Purpose** | Verify that the Client handles out of order delivery as well the right replacement on received SIs in a proper way. |
| **Specification Reference** | [PushSI] Section 8.2.1.1<br>[PushSI] Section 6.2<br>[PushSI] Section 7<br>[PushSI] Section 8<br>[PushSI] Section 6.4 |
| **SCR Reference** | SI-CF-C-005<br>SI-SEM-C-003, SI-SEM-C-004, SI-SEM-C-005 and SI-SEM-C-006 Section 6.2.<br>SI-CF-C-001<br>SI-CF-C004<br>SI-SEM-C-015, SI-SEM-C-016 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test procedure** | Send an SI to the Client.<br>Send a second SI with the same SI-ID and newer CREATED attribute value than in the first one.<br>Send a third SI with the same SI-ID and older CREATED attribute value than in the second SI. |
| **Pass-Criteria** | The Client must replace the older (i.e. the first to be sent) SI by the received newer one.<br><br>The Client must silently discard the received older one. |
| **Comment** | None. |

## 6.4.4 One or Multiple SIs that are not processed upon reception

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-26 |
| **Test Object** | Client device |
| **Test Case Description and Purpose** | Verify that received SI messages are stored in the Inbox according to the Clients implementation and the ability to maintain at least one SI that cannot be processed directly upon reception |
| **Specification Reference** | [PushSI] Section 8.2.1.1<br>[PushSI] Section 6.2.1<br>[PushSI] Section 6.2<br>[PushSI] Section 7<br>[PushSI] Section 8<br>[PushSI] Section 6.4 |
| **SCR Reference** | SI-CF-C-005<br>SI-SEM-C-007 and SI-SEM-C-008<br>SI-SEM-C-009 and SI-SEM-C-012<br>SI-SEM-C-011<br>SI-CF-C-001<br>SI-CF-C004<br>SI-SEM-C-015, SI-SEM-C-016 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either “Always ask”, “Always” or “Never”.<br>Current date / time are set on the MS.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send various SIs with different ACTION attributes (“Execute-Low”, “Execute-Medium”, “Execute-High”) and with different creation time. |
| **Pass-Criteria** | The Push Inbox list is sorted according to the following logic.<br>1. The Action attribute (High Medium Low)<br>2. The Order that they are received by the device.<br>The Client is able to maintain at least one message that is not directly processed. |
| **Comment** | Sort order shall be dependent on:<br>• action attribute<br>• Order of reception<br>• loaded/read or unloaded/unread push messages |

## 6.4.5 Push Accessibility User Settings

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-27 |
| **Test Object** | Client Device |
| **Test Case Description and Purpose** | Ability to choose the load the service immediately or postpone the SI for later handling<br>Verify that when an SI is presented, the end-user can choose to load the service immediately, or postpone the SI for later handling depending on user settings in Push Accessibility.<br>Verify that the Client provides the end-user the ability to act on the postponed message.<br>Verify the ability to maintain at least one postponed message |
| **Specification Reference** | [PushSI] Section 6.3<br>[PushSI] Section 6.3.1<br>[PushSI] Section 6.4<br>[PushSI] Section 7<br>[PushSI] Section 8 |
| **SCR Reference** | SI-SEM-C-009 and SI-SEM-C-010<br>SI-SEM-C-013 and SI-SEM-C-014<br>SI-SEM-C-016, SI-SEM-C-015, SI-SEM-C-016<br>SI-CF-C-001 Section 7.<br>SI-CF-C004 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either “Always ask”, “Always” or “Never”.<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an SI message to the Client. |
| **Pass-Criteria** | The Client should be able to load the SI & SL messages.<br><br>When the Push accessibility settings are set to “Always ask”, the end- user will always be asked if the message will be processed now or later.<br><br>If accessibility is set to “Always”, the user will not be asked but the message will be processed directly.<br><br>If the accessibility setting is set to “Never”, then no Push messages will be accepted by the Client.<br><br>When the end-user would like to processes the postponed message, the Client should provide the end-user the ability to do so.<br><br>The end-user can abort the service. |
| **Comment** | None |

# 6.5 Server/Client Service Loading

## 6.5.1 Character Encoding

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-28 |
| **Test Object** | Handling of the supported character encodings |
| **Test Case Description and Purpose** | Verify that the received SL messages with the supported encodings are processed by the Client in the correct way. |
| **Specification Reference** | [PushSL] Section 5.1<br>[PushSL] Section 8<br>[PushSL] Section 9 |
| **SCR Reference** | SL-CSE-C-005, SL-CSE-C-006 and SL-CSE-C-007<br>SL-CF-C-001<br>SL-CF-C-003<br>SL-PPG-S-001<br>SI-PPG-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either “Always ask”, “Always” or “Never”.<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send some SLs with the supported encoded characters to the Client. |
| **Pass-Criteria** | The Client should be able to load the SL messages with the supported encoded characters.<br><br>Client must be able to ignore messages with unsupported encoded characters. |
| **Comment** | None |

## 6.5.2 One or multiple SL messages that are not processed upon reception

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-29 |
| **Test Object** | Handling of multiple SLs that are not processed upon reception |
| **Test Case Description and Purpose** | Verify that received SL messages are stored in the Inbox according to the Clients implementation and the ability to maintain at least one SL that cannot be processed directly upon reception. |
| **Specification Reference** | [PushSL] Section 6.1<br>[PushSL] Section 6.2<br>[PushSL] Section 8<br>[PushSL] Section 9 |
| **SCR Reference** | SL-SEM-C-001<br>SL-SEM-C-003<br>SL-CF-C-001<br>SL-CF-C-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set either to “Always ask”, “Always” or “Never”.<br>Current date / time are set in the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send various SLs with different ACTION attribute (“Execute-Low”, “Execute-Medium”, “Execute-High”) and with different creation time. |
| **Pass-Criteria** | The Push Inbox list is sorted according to the following logic.<br><br>1. The Action attribute (High Medium Low)<br><br>2. The Order that they are received by the device |
| **Comment** | Sort order shall be dependent on:<br>• action attribute<br>• The order of reception of the messages<br>• loaded/read or unloaded/unread push messages |

## 6.5.3 Push Accessibility User Settings

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-30 |
| **Test Object** | Client device |
| **Test Case Description and Purpose** | Ability to choose to load the service immediately or postpone the SL for later handling:<br><br>Verify that when an SL is presented, the end-user can choose to load the service immediately, or postpone the SL for later handling depending on user settings in Push Accessibility.<br>Verify that the Client provides the end-user the ability to act on the postponed message.<br>Verify the ability to maintain at least one postponed message |
| **Specification Reference** | [PushSL] Section 6.2<br>[PushSL] Section 8<br>[PushSL] Section 9 |
| **SCR Reference** | SL-SEM-C-003<br>SL-CF-C-001<br>SL-CF-C-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user setting is set to either “Always ask”, “Always” or “Never”.<br>Current date / time are set in the Client.<br>The right PPG IP address is set I the Clients currently activeWAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send an SL message to the Client. |
| **Pass-Criteria** | When the Push accessibility settings are set to “Always ask”, the end- user will always be asked if the message will be processed now or later.<br><br>If accessibility is set to “Always”, the end-user will not be asked but the message will be processed directly.<br><br>If the accessibility setting is set to “Never”, then no Push messages will be accepted by the Client.<br><br>When the end-user would like to processes the postponed message, the Client should provide the end-user the ability to do so.<br><br>The end-user can abort the service. |
| **Comment** | None |

# 6.6 HTTP Server/Client Connections

## 6.6.1 Unsecure (TO-TCP)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-31 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | To verify that a Terminal oriented Unsecure Method allows the Push message to be sent by the server and received by the client when it is on idle mode or during an ongoing WAP connection. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.4.1 |
| **SCR Reference** | OTA-HTTP-S-001 (OTA-CO-S-003)<br>OTA-HTTP-C-001 (OTA-CO-C-003) Req: TCP:MCF |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The device<br>Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The Client will connect via HTTP to the Server via (in defined order of precedence):<br><br>1 .A specified port defined in the SIR (If applicable WSP – 2948 Http - 4035)<br>2. A Provisioned port (If Applicable)<br>3. One or more registered Push ports (Non Secure/ Secure)<br><br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR, then the client should establish a push session towards one of the contact points. It is left to the device to decide which protocol variant to use.<br><br>However the SIR may indicate that it accepts any Application ID. Therefore the client has the responsibility to clean up the stale push sessions |
| **Pass-Criteria** | The client must establish an IP connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-HTTP.<br><br>Retrieve the Push messages that have been queued for the Client.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |

| | |
|---|---|
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing HTTP connection. Should also be executed using both SI and SL messages |

## 6.6.2 Secure (TO-TCP)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-32 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | To verify that a terminal oriented Secure TLS Method allows the Push message to be sent by the server and received by the client when it is on idle mode or during an ongoing WAP connection. |
| **Specification Reference** | [PushOTA] Section 7.2.4.1 |
| **SCR Reference** | OTA-HTTP-S-003 (Reg: TLS:MSF)<br>OTA-HTTP-C-003 (Reg: TLS:MCF) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The Client will connect via HTTP-TLS to the Server via (in defined order of precedence):<br><br>1. A specified port defined in the SIR (If applicable)<br>2. A Provisioned port (If Applicable)<br>3. One or more registered Push ports (Non Secure/ Secure)<br><br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR, then the client should establish a push session towards one of the contact points. It is left to the device to decide which protocol variant to use. |
| **Pass-Criteria** | The client must establish an IP connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-HTTP-TLS.<br><br>Retrieve the Push messages that have been queued for the Client.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing HTTP connection. Should also be executed using both SI and SL messages |

## 6.6.3 Unsecure (PO-TCP)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-33 |
| **Test Object** | Server Client |
| **Test Case Description and Purpose** | To verify that a PPG oriented UNSecure Method allows the Push message to be sent by the server and received by the client when it is on idle mode or during an ongoing WAP connection. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.4.2 |
| **SCR Reference** | OTA-HTTP-S-002<br>OTA-HTTP-C-002 (OTA-CO-C-003) Req: TCP:MSF |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PPG Server will send an SI and SL Push messages to the client.<br><br>The Client will connect via HTTP to the Server via (in defined order of precedence):<br><br>1. A specified port defined in the SIR (If applicable)<br>2. A Provisioned port (If Applicable)<br>3. One or more registered Push ports (Non Secure/ Secure)<br><br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR , then the client should establish a push session towards one of the contact points. It is left to the device to decide which protocol variant to use.<br><br>However the SIR may indicate that it accepts any Application ID. Therefore the client has the responsibility to clean up the stale push sessions. |
| **Pass-Criteria** | The client must establish an IP connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-HTTP.<br><br>Retrieve the Push messages that have been queued for the Client.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing HTTP connection.<br>Should also be executed using both SI and SL messages |

## 6.6.4 Secure (PO-TCP)

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-34 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | To verify that a PPG oriented Secure Method allows the Push message to be sent by the server and received by the client when it is on idle mode or during an ongoing WAP connection. |
| **Specification Reference** | [PushOTA] Section 7.2.4.1.2 |
| **SCR Reference** | OTA-HTTP-S-004 Req: TLS:MSF<br>OTA-HTTP-C-004 Req: TLS:MCF |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PPG Server will send an SI and SL Push messages to the client.<br><br>The Client will connect via HTTP-TLS to the Server via (in defined order of precedence):<br><br>1. A specified port defined in the SIR (If applicable)<br>2. A Provisioned port (If Applicable)<br>3. One or more registered Push ports (Non Secure/ Secure)<br><br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR , then the client should establish a push session towards one of the contact points. It is left to the device to decide which protocol variant to use.<br><br>However the SIR may indicate that it accepts any Application ID. Therefore the client has the responsibility to manage push sessions. |
| **Pass-Criteria** | The client must establish an IP connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-HTTP-TLS.<br><br>Retrieve the Push messages that have been queued for the Client.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comments** | This test case should be executed under all conditions of the Push access user settings when the Client is both in idle and during an ongoing HTTP connection. |

| | |
|---|---|
| | Should also be executed using both SI and SL messages |

## 6.6.5 Registration

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-35 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that PPG initiated Registration is accomplished by sending an HTTP-OPTIONS [RFC 2626] request to the client using a push message which contains a Request URI and an empty Host header field. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.5.1<br>[PushOTA] Section 7.2.5 |
| **SCR Reference** | OTA-HTTP-S-005<br>OTA-HTTP-C-005 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The X-Wap-Push-ProvURL header may be included in the request.<br>The response from the Client unless rejection occurs must include the following headers if no X-WAP-CPITag header is conveyed:<br><br>1. CPI headers(Optional headers specified)<br>2. The X-Wap-CPITag header<br><br>The above headers are also included in the response if a CPITag is conveyed from the PPG and doesn't match the Current Clients CPITag<br><br>The CPITag is calculated by base64 encoding of the four octets and can be conveyed to the Client by the PPG in either method:<br><br>1. Include the CPITag in an SIR<br>2. Include the CPITag in the X-Wap-CPITag header in the Options request<br><br>Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set it in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send the HTTP-OPTIONS request to the Client.<br><br>The client will respond to the OPTIONS request with a response status code that reflects the outcome of that request i.e. (accepted, authentication required ...) The X-WAP-Push-Status header indicating the outcome of the registration request must be included in the response to the OPTIONS request. |
| **Pass-Criteria** | The Client must accept and process the HTTP-OPTIONS Request<br><br>The Response code: Accepted & X-WAP-Push-Status: 500 if CPITag matches |

| | |
|---|---|
| | The Response code: NOT Accepted & X-WAP-Push-Status: 501 if CPITag does not match. |
| **Comment** | Response Status Codes:<br><br>234-299 – Push Request Rejected<br>300-399 – Registration Request Rejected<br>400-499 – Push request accepted<br>500-599 – Registration request accepted<br>600-699 – General Rejections Reasons |

### 6.6.6 Registration Validation

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-36 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify and validate that the PPG's CPI context is current and matched as to the one on the Client. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.5<br>[PushOTA] Section 7.2.5.2 |
| **SCR Reference** | OTA-HTTP-S-006<br>OTA-HTTP-C-006 (OTA-CO-C-003) |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | This method only works when the PPG knows the coupling between Identity and IP address of the Clients current CPI.<br>The CPITag<br>Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile. The Client should be in the standby mode.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | If the CPITag assumed by the PPG matches the Clients current CPITag then the client will process the push message. Thus respond by NOT including the X-Wap-CPITag header in the post response.<br><br>If the CPITag does not match the Clients then the client will silently disregard the message. Then the Client conveys its X-Wap-CPITag header in the response |
| **Pass-Criteria:** | The Client should accept and process the messages accordingly. |
| **Comment** | None. |

## 6.6.7 CPI and User Agent Profile

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-37 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify and validate that the PPG's can use the CPI context or the **X-wap-profile** and **X-wap-profile-diff** headers if available. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.5.6 |
| **SCR Reference** | |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | This method will validate the use of a UAprofile. The Push attributes of the UAprofile will be referenced when establishing the Clients CPI However if the CPI has been defined in the Options request . Then the Uaprof should NOT supersede these definitions.<br><br>Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile. The Client should be in the standby mode.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Verify that the defined Push capabilities of the Client are correctly matched to the referenced UAprofile Push characteristics for the device. Failing this then the capabilities should be referenced from the CPI if defined. |
| **Pass-Criteria:** | The Client should accept and process the messages accordingly to the devices defined capabilities. |
| **Comment** | None. |

## 6.6.8 Un-Authenticated Terminal Identification

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-38 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Client is UN authenticated by the PPG and fails Message delivery. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.6.1 |
| **SCR Reference** | OTA-HTTP-S-007 (OTA-CO-S-003)<br>OTA-HTTP-C-007 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Send the HTTP-OPTIONS request to the Client.<br><br>The client will respond to the OPTIONS request with a response in ABNF [RFC2234] format. |
| **Test Procedure** | The PPG will send the HTTP-OPTIONS with an ***X-Wap-Authenticate header*** request to the Client.<br><br>The client will respond to the OPTIONS request with an ***X-Wap-Authorization header*** containing the Terminal-ID if it accepts the challenge.<br><br>The Client must NOT include the X-Wap-Authorization header in the response unless the X-Wap-Authenticate header was present in the corresponding request. |
| **Pass-Criteria** | The Client must reject the message upon failed Authentication |
| **Comment** | |

## 6.6.9 Authenticated Terminal Identification

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-39 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Client can be authenticated by the PPG when requested to do so. |
| **Specification Reference** | [PushOTA] Section 7.<br>[PushOTA] Section 7.2.6.2 |
| **SCR Reference** | OTA-HTTP-S-008 (OTA-CO-S-003)<br>OTA-HTTP-C-008 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Send the HTTP-OPTIONS request to the Client.<br><br>The client will respond to the OPTIONS request with a response in basic or digest [RFC2617] format. Both Client and Server MUST support basic and MAY support digest authentication scheme |
| **Test Procedure** | The PPG will send the HTTP-OPTIONS with an ***X-Wap-Authenticate header*** request to the Client.<br><br>The client will respond to the OPTIONS request with an ***X-Wap-Authorization header*** containing the Terminal-ID if it accepts the challenge.<br><br>The client will respond to the OPTIONS request with a 412 “precondition Failed” and include the auth-parm directive in the X-Wap-Authorization header if it rejects the challenge.<br><br>The Client must NOT include the X-Wap-Authorization header in the response unless the X-Wap-Authenticate header was present in the corresponding request. |
| **Pass-Criteria:** | The Client should accept and process the messages when Authorisation . |
| **Comment** | None. |

## 6.6.10 Authenticated PPG Identification

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-40 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the PPG Server can be authenticated by the Client when requested to do so. Also that the terminal can handle Non Authentication PPG ID |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.6.2 |
| **SCR Reference** | OTA-HTTP-S-009 (OTA-CO-S-003)<br>OTA-HTTP-C-008 (OTA-CO-C-003)<br>OTA-HTTP-C-010 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Send the WWW-Authenticate request to the Server.<br><br>The Server will respond to the WWW-Authenticate header request with a response in basic or digest [RFC2617] format. Both Client and Server MUST support basic and MAY support digest authentication scheme |
| **Test Procedure** | The Client will send the HTTP-OPTIONS request with a WWW-Authenticate header request to the Server.<br><br>The Server will respond to the WWW-Authenticate request with an X-Wap-Authorization header containing the following if it accepts the challenge.<br><br>Realm = Terminal-ID<br>domain = /wappush<br>username = Proxy-ID<br>stale = NA<br>Alogorithm = SHA-1<br>qop-options = NA<br>nonce= Unique generated number<br><br>The client can respond to the OPTIONS request with a 401 “Unauthorized” to the PPG so that the PPG resend its authorization details. |
| **Pass-Criteria** | The Server must accept and process the messages |
| **Comment** | None |

## 6.6.11 Application Addressing

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-41 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Server addresses the appropriate Client push application with the absolute path as the URI of the Post Request. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.3 |
| **SCR Reference** | OTA-HTTP-S-010 (OTA-CO-S-003)<br>OTA-HTTP-C-011 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The message body of the POST request uses /wappush as the Request URI and an empty Host header field then the remaining content and headers for the addressed application. Upon response the Status code is returned 400 Accepted. |
| **Test Procedure** | Send a Push request from the Server with /wappush as the URI of the Post to the Client. |
| **Pass-Criteria** | The Client must accept and process the message by the appropriate application I.e. MMS, WapPush, DRM. The received Application PUSH can be presented successfully.<br><br>In the UDP logs verify that the last push message flag is set and that the process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | The procedure should be executed for connectionless, connection oriented Push Service Indication, Service Loading and Session Initiation Application messages when the Client is on standby as well as during an ongoing WAP connection. |

## 6.6.12 Content Push

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-42 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Server sends the following Push content to the Client in the proper acceptable format. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.4 |
| **SCR Reference** | OTA-HTTP-S-011 (OTA-CO-S-003)<br>OTA-HTTP-C-012 (OTA-CO-C-003) |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | |
| **Test Procedure** | The Push message body of the Post can include the following Headers:<br><br>X-Wap-Push-Info – This is a request header used in the Post request sent by the Ppg to provide the terminal with the following indications (Authenticated, Trusted, Last, response)<br><br>X-Wap-Push-ProvURL – This header needs to be included in the first HTTP request sent to the terminal using PO-TCP method. This then allows the client to associate that active TCP connection with a certain ProvURL until the connection is closed. (This is only applicable to Clients supporting Wap provisioning) |
| **Pass-Criteria:** | The Client should accept and process the messages. |
| **Comment** | None. |

### 6.6.13 Version Control

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-43 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Client can handle Version Control. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.5 |
| **SCR Reference** | OTA-HTTP-S-012 (OTA-CO-S-003)<br>OTA-HTTP-C-013 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The OTA protocol over HTTP allows version control of major minor integer values releases.<br><br>A*star can be used in the minor integer value to indicate acceptance for all minor versions of a given major release. |
| **Test Procedure** | The X-Wap-Push-OTA-Version header is included in the first HTTP response; if it is included in the first request then it is required to be in the response.<br><br>If the client is not able to handle any of the versions indicated by the PPG, the client MUST include the X-Wap-Push –Status header with a value of 600 and an appropriate text message ”Version Not Supported” in the response. |
| **Pass-Criteria** | The version is communicated successfully |
| **Comment** | None. |

## 6.6.14 Security Considerations

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-44 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | To protect the device against Denial of Service attacks the client should implement a LOCKOUT timer. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 8.3 |
| **SCR Reference** | OTA-HTTP-S-013<br>OTA-HTTP-C-014<br>OTA –CO-C-003)<br>OTA-SEC-C-006 AND<br>OTA-SEC-C-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The lockout timer is reset if the requested push session is successfully established (OTA-WSP) or (OTA-HTTP)<br>If the SIR is sent via a secure port then the Security measures do not have to be necessary.<br>To protect against spoofing the Client should validate the SIR by comparing source address of PDU that carries SIA content with the set of prestored ones. |
| **Test Procedure** | Test Denial of service by sending a Push to the client and then send further wappushs during the lockout period.<br>During Lockout period the notifications are disregarded until the timer expires which is device specific. Upon which the client resumes to normal operation. |
| **Pass-Criteria** | The Client should accept and process the messages. |
| **Comment** | |

## 6.6.15 Bearer Indication

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-45 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Client can register different bearers with the PPG. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.6 |
| **SCR Reference** | OTA-HTTP-S-014 (OTA-CO-S-003)<br>OTA-HTTP-C-015 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The Bearer type is defined in RFC2234 format. |
| **Test Procedure** | The PPG will send the HTTP-OPTIONS request to Client.<br><br>The Client will respond with an X-Wap-Bearer-Indication header containing the following Bearer type. |
| **Pass-Criteria** | The Client must accept and process the messages on the bearer defined. |
| **Comment** | |

## 6.6.16 SIA/SIR

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-46 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | To verify that the server sends the appropriate SIR to the Client and the SIA in the client will service the request. |
| **Specification Reference** | [PushOTA] Section 8<br>[PushOTA] Section 8.1<br>[PushOTA] Section 8.4 |
| **SCR Reference** | OTA-HTTP-S-015 (OTA-C-S-003)<br>OTA-HTTP-C-016 (OTA-CO-C-003)<br>OTA-SEC-C-006 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The device<br>Device Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs.<br><br>If the secure session HTTP-TLS is requested by using the SIR secure port or a provisioned port then the client must ensure that a HTTP-TLS session exists before it creates a new push session. |
| **Test Procedure** | The PPG Server will send an SIR Push message to the client.<br>If multiple contact points (OTA WSP / OTA HTTP) are included in the SIR, then the client should establish a push session towards one of the contact points. It is left to the device to decide which protocol variant to use.<br>However the SIR may indicate that it accepts any Application ID. Therefore the client has the responsibility to clean up the stale push sessions. |
| **Pass-Criteria:** | The Client must accept the SIR and process the message by the SIA and the application ID. The client will carry out the following<br><br>The client must establish a connection to the network, if not already done so.<br><br>Establish push sessions towards the contact points via OTA-HTTP defined in the SIR.<br><br>In the UDP logs verify that the last push message flag is set and that process has been completed successfully<br><br>The Server logs can be monitored to confirm that acknowledgment of delivery has been complete. |
| **Comment** | None. |

## 6.6.17 Support for the X-Wap-Push-ProvURL header

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-47 |
| **Test Object** | Server/Client |
| **Test Case Description and Purpose** | Verify that the Client handles a push message which contains an X-Wap-Push-ProvURL header set to a proper value in a proper way. |
| **Specification Reference** | [PushOTA] Section 7<br>[PushOTA] Section 7.2.5.4 |
| **SCR Reference** | OTA-HTTP-S-016<br>OTA-HTTP-C-017 (OTA-CO-C-003) |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | If the specified Provisioning URL is non empty and it matches the terminals config then the URL is used.<br><br>If the specified Provisioning URL is NOT matching the terminals config then the Terminals URL is used, and a return status code is returned (257 or 302) in the X-Wap-Push-Status header.<br><br>If the specified Provisioning URL is empty it is the discretion of the terminal to select the appropriate config context among those having an empty ProvURL.<br><br>If the terminal cannot find a Provisioning context with an empty ProvURL the request is rejected and the return status code is returned (257 or 302) in the X-Wap-Push-Status header.<br><br>If the specified Provisioning URL is present and the terminal does not support WAP provisioning the terminal may reject the request and a return status code is returned (257 or 302) in the X-Wap-Push-Status header. |
| **Test Procedure** | X-Wap-Push-ProvURL – This header needs to be included in the first HTTP request sent to the terminal from the PPG using PO-TCP method. This then allows the client to associate that active TCP connection with a certain ProvURL until the connection is closed. (This is only applicable to Clients supporting Wap provisioning) |
| **Pass-Criteria** | The Client must accept and process the messages |
| **Comment** | |

# 6.7 Push Message

## 6.7.1 Content-Type header

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-48 |
| **Test Object** | Client device |
| **Test Description and Purpose** | Verify that the Client handles push messages with different content type headers in a proper way. |
| **Specification Reference** | [PushMsg] Section 5.2.1.10<br>[PushMsg] Section 5.2.1<br>[PushMsg] Section 5.2.3<br>[PushMsg] Section 5.3<br>[PushMsg] Section 6<br>[PushSI] Section 7<br>[PushSI] Section 8<br>[PushSL] Section 8<br>[PushSL] Section 9 |
| **SCR Reference** | MSG-GEN-C-002<br>MSG-GEN-C-001<br>MSG-GEN-C-003<br>MSG-GEN-C-004<br>MSG-GEN-C-005, MSG-GEN-C-006 and MSG-GEN-C-007<br>MSG-GEN-S-002<br>MSG-GEN-S-001<br>MSG-GEN-S-003<br>MSG-GEN-S-004<br>MSG-GEN-S-005 (Req: MSG-GEN-C-006 )<br>MSG-GEN.S-007<br>MSG-GEN-S-008<br>SI-CF-C-001<br>SI-CF-C004<br>SL-CF-C-001<br>SL-CF-C-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval.<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG. |
| **Test Procedure** | Send push messages to the Client. |
| **Pass-Criteria** | The Client should be able to process messages with trusted header types and be able to discard messages with untrusted header types. |
| **Comment** | Any type of message is applicable here i.e. either Service Loading or Service Indication and when the Client is both on standby as well as during an ongoing WAP or HTTP connection |

## 6.7.2 Support for 4 Concatenated SMS's

| | |
|---|---|
| **Test Case ID** | Push-OTA-2.2-int-49 |
| **Test Object** | Server/client device |
| **Test Case Description and Purpose** | Verify that the Server can support concatenating 4 segmented SMS messages for a large payload. |
| **Specification Reference** | [PushOTA] Section 6.2<br>[PushOTA] Section 6.2.1.1 |
| **SCR Reference** | WDP-CDMA_C-001 OR<br>WDP-GSM-C-001<br>WDP-CDMA_S-001 OR<br>WDP-GSM-S-001 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Client Push inbox and cache content are empty.<br>Push access user settings are set to allow Push with automatic retrieval<br>Current date / time are set on the Client.<br>The right PPG IP address is set in the Clients currently active WAP Profile.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the Client and the PPG.<br>Recommended to use UDP logs. |
| **Test Procedure** | The PI will send an large Un Confirmed Push PAP message to the PPG<br>The PPG Server will then segment the SI or SL Push messages to the client via 4 SMS messages to transmit the large payload. The Client will then process the Push message by concatenating the push messages. |
| **Pass-Criteria** | The client will be concatenating the messages to formulate the completed message payload. The client will also validate the push message source address against the Whitelist. If successful the push message will be processed otherwise it will be ignored. If no whitelist is defined then by default the Push will be accepted. |
| **Comments** | |

# 7. PAP Test Cases

There are 24 interoperability test cases for WAP 2.1 Push Enabler.

## 7.1 Validation of XML Push Initiator

### 7.1.1 Validate XML in control Entity in Push Submission

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-1 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML control entity in the Push submission by the initiator to the PPG. |
| **Specification Reference** | [OMA-WAP-TS-PAP] Section 5.1 |
| **SCR Reference** | PAP-VAL-S-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and receiptant Device MSISDN are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with an Error Code defined in Appendix A depending upon the error condition during validation of the data contained in the Message.<br>The PPG will then handle the successfully delivery of the message to the receiptant Device. |
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission. |
| **Comment** | |

## 7.1.2 Validation of content Entity

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-2 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML content entity in the Push submission by the initiator to the PPG. |
| **Specification Reference** | [OMA-WAP-TS-PAP] Section 5.1 |
| **SCR Reference** | PAP-VAL-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and receiptant Device MSISDN are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with an Error Code defined in Appendix A depending upon the error condition during validation of the data contained in the Message.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device. |
| **Pass -Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission. |
| **Comment** | |

## 7.1.3 Validation of Addresses

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-3 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Receiptant Address is submitted by the initiator to the PPG. |
| **Specification Reference** | [OMA-WAP-TS-PAP] Section 6.1 |
| **SCR Reference** | PAP-VAL-S-003 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN's or addresses are pre configured on the PPG.<br><br>There are three types of addresses 1. The push proxy gateway Address, 2. The wireless Device address. 3. Result notification address<br><br>The Wireless Device address is the considered entity here and can have many formats i.e. an IP address or MSISDN / Subscriber number.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receiptants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response (***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command ( ***statusquery-message***) can be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices.<br><br>The receiptant's address has to exist otherwise the message will not be sent as the user is unknown. |
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission. |
| **Comments** | PPG/Initiator |

# 7.2 Operations

## 7.2.1 Push Submission

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-4 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Receiptant Address is added in the Push submission by the initiator to the PPG. |
| **Specification Reference** | [OMA-WAP-TS-PAP] Section 6.1 |
| **SCR Reference** | PAP-OPS-S-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>When the PPG returns the Push-Response message after a push submission to multiple recipients, the response corresponds to the message submission and not the number of receiptants so only one response per message submission.<br><br>Result notifications are returned by the PPG per receiptant if the result notification is requested the PI during the submission of a message.<br><br>If a cancel message is submitted the PPG may send back individual responses related to each receiptant's message or a response status for all the messages<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message or Messages sent to one or more receipt ants on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with a message response depending upon the validation of the data contained in the Message.<br><br>Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response (***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command (statusquery***-message***) should be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |

| | |
|---|---|
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission. The message will then be placed on the Push Queue for Delivery or it will be delivered immediately to the device. |
| **Comment** | |

### 7.2.2 Result Notification Response

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-5 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Result Notification response is sent to the Initiator. This will communicate the final outcome of the push message; it may also contain the content entity from the wireless device. |
| **Specification Reference** | [OMA-WAP-TS-PAP] Section 6.1 |
| **SCR Reference** | PAP-OPS-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response (depending upon the validation of the data contained in the Message.<br>The PI should understand the message response and upon its message type make a pass or error condition<br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |
| **Pass-Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful. |
| **Comment** | |

### 7.2.3 Push Cancellation

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-6 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML Push Cancellation command can be sent to the PPG and |

| | |
|---|---|
| | the PPG respond to the command appropriately. |
| **Specification Reference** | [Push WAP247-PAP] Section 5.3 |
| **SCR Reference** | PAP-OPS-S-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response (***resultnotification-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Cancellation command ( ***cancel-message***) should be sent to the PPG with the Push ID or PushID plus message ID This should be sent after the initial submission so allowing the PPG time to cancel the pending push.<br>The PPG will respond with a Cancel Response(***cancel-response***) status<br>The PPG will then handle cancellation of the message if the message hasn't already been sent. |
| **Pass -Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful.<br>The PPG responds to a cancellation command ( ***cancel-message***) submitted by the initiator with the (***cancel-response)*** Message.<br>If the PPG does not support the cancellation functionality when a cancellation command ( ***cancel-message***)is submitted by the initiator the (***cancel-response) NOT Implemented*** MUST be returned to Initiator. |
| **Comment** | |

## 7.2.4 Status Query

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-7 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML Push Status command can be sent to the PPG and the PPG respond to the command appropriately. |

| | |
|---|---|
| **Specification Reference** | [Push WAP247-PAP] Section 5.4 |
| **SCR Reference** | PAP-OPS-S-004 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message. Once a message has been submitted then a Status command ( ***statusquery-message***) should be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br>The PPG will then handle the successful delivery and cancellation of the message |
| **Pass -Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful.<br>If the PPG does not support the status functionality when a status command is submitted by the initiator, then the response NOT ***Implemented*** MUST be returned to Initiator.<br>The PPG will then handle the successful delivery and command validation |
| **Comment** | |

## 7.2.5 Client Capabilities Query

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-8 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML Push Capabilities Query command can be sent to the PPG and the PPG respond to the command appropriately. |
| **Specification Reference** | [Push WAP247-PAP] Section 7.3 |
| **SCR Reference** | PAP-OPS-S-005 |
| **Tool** | PUSH INITIATOR |

| | |
|---|---|
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Client Capabilities Query command ( ***ccq-message***) should be sent to the PPG .<br>The PPG will then handle the successful delivery and query of the message |
| **Pass -Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful.<br>If the PPG does support the client capabilities functionality when a (***ccq-mesaage)*** command is submitted by the initiator, then the (***ccq response***) XML doc is returned to Initiator containing the PPG and the Devices capabilities as defined from the UAProfile.<br>If the PPG does NOT support the client capabilities functionality when a (***ccq-mesaage)*** command is submitted by the initiator, then the (***ccq response***) XML doc is returned to Initiator containing the status " NOT IMPLEMENTED" |
| **Comment** | |

# 7.3 Push Semantics

## 7.3.1 Support for multiple recipient addresses

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-9 |
| **Test Object:** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing multiple receipt ants and the PPG respond to the command appropriately. |
| **Specification Reference** | [Push WAP247-PAP] Section 6.1 |
| **SCR Reference** | PAP-SEM-S-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | |
| **Pass-Criteria** | Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response(***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command ( ***statusquery-message***) can be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |
| **Comment** | Note Multiple recipients is an Optional feature of PPG |

## 7.3.2 Support for multiple addresses in responses

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-10 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing multiple receipt ants and the PPG respond to the command appropriately. |
| **Specification Reference** | [Push WAP247-PAP] Section 6.1 |
| **Requirement:** | PAP-SEM-S-002 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedures** | Compose a Message on the Push Initiator and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status Query command ( ***statusquery-message***) can be sent to the PPG . This will result in a separate status response to be returned to the Initiator for each receiptant on the multiple message submission If a query is needed for each receiptant message per submission then a result notification is requested during submission<br>The PPG will then handle the successful delivery and query of the message |
| **Pass-Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful.<br>If the PPG does support the status query functionality when a (***statusquery-message)*** command is submitted by the initiator, then the (***statusquery response***) XML doc is returned to Initiator containing the status of each receipt ants message status.<br>If the PPG does not support the status functionality when a status command is submitted by the initiator, then the ccq response ***NOT Implemented*** MUST be returned to the Initiator.<br>The PPG will then handle the successful delivery and command validation |
| **Comment** | Note Multiple recipients is an Optional feature of PPG |

## 7.3.3 Deliver after Time stamp

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-11 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***deliver-after timestamp*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2 |
| **SCR Reference** | PAP-SEM-S-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Prerequisites:** | The attribute Deliver after timestamp specifies the time and date by which the content must be delivered to the wireless device. Content that has aged beyond this date must be transmitted. The date format must be in Co-ordinated Universal Time (UTC)<br>The PPG MUST reject the message submission from the initiator if the PPG does not support this function |
| **Test procedure:** | Compose a Message on the Push Initiator and send it with various Deliver after date and time stamps to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>Create a message with a Deliver after date time stamp greater and less than present time<br>The PPG will then handle the successful delivery as the time stamp specifies |
| **Pass criteria:** | Ensure that the message that has a timestamp Less than present time is ignored and fails to deliver as the time has expired. The greater time stamp should deliver correctly after the expiry time has an elapsed in respect to current date and time as specified in the message attribute. |
| **Comment** | |

## 7.3.4 Deliver Before Time stamp

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-12 |
| **Test object** | PPG/Initiator**.** |
| **Test case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***deliver-before timestamp*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2 |
| **SCR Reference** | PAP-SEM-S-004 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute Deliver before timestamp specifies the time and date by which the content must be delivered to the wireless device. Content that has aged beyond this date must be transmitted. The date format must be in Co-ordinated Universal Time (UTC)<br>The PPG MUST reject the message submission from the initiator if the PPG does not support this function |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with various Deliver before date and time stamps to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>Create a message with a Deliver before date time stamp greater and less than present time<br>The PPG will then handle the successful delivery as the time stamp specifies |
| **Pass-Criteria** | Ensure that the message that has a timestamp Less than present time is accepted and successfully delivers as this time has expired. The greater time stamp should deliver correctly before the expiry time is reached as specified in the message attribute. |
| **Comment** | None |

## 7.3.5 Failed Requests when QOS cannot be honoured

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-13 |
| **Test Object** | PPG/Initiator. |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***quality-of-service*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2.2 |
| **Requirement:** | PAP-SEM-S-005 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***Quality-of-Service*** specifies the quality of Service by which the content must be delivered to the wireless device<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with various QOS types' elements to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>Create a message with a QOS type that is not supported or a mis-spelling of the types<br><br>The PPG will then handle the successful delivery if the QOS is accepted. If the QOS is rejected then an error code is returned as specified in the Appendix A Error Status Codes<br>Reference of Types:<br>Priority<br>Delivery Method<br>Network<br>Bearer |
| **Pass-Criteria** | Ensure that the correct Error Status code as defined in Appendix A is used in the response to incorrect or unsupported QOS. |
| **Comment** | None |

## 7.3.6 Delivery method in QOS

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-14 |
| **Test Object** | PPG/Initiator**.** |
| **Test Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***deliver-method*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2.2 |
| **SCR Reference** | PAP-SEM-S-006 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***Quality-of-Service*** specifies the quality of Service by which the content must be delivered to the wireless device<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with various Delivery Methods<br>Confirmed, Prefer confirmed, Confirmed-with-response, One-shot, Unconfirmed, Not specified.<br><br>The PPG will use this Delivery method to decide the over the air delivery method desired by the Push initiator<br><br>The PPG will then handle the successful delivery if the QOS is accepted. If the QOS is rejected then an error code is returned as specified in the Appendix A Error Status Codes |
| **Pass-Criteria** | Ensure that the correct Error Status code as defined in Appendix A is used in the response to incorrect or unsupported QOS. |
| **Comment** | None. |

## 7.3.7 Priority delivery

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-15 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***priority*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2.2 |
| **SCR Reference** | PAP-SEM-S-007 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***priority*** specifies the delivery priority of the message. Valid values are " LOW MEDIUM HIGH" by which the order of the message must be delivered to the wireless device<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test procedure** | Compose a Message on the Push Initiator and send it with various priority types to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>Create a message to the same receiptant with a LOW MEDIUM HIGH priority<br><br>The PPG will then handle the successful delivery if the priority is accepted. If the priority is rejected then an error code is returned as specified in the Appendix A Error Status Codes |
| **Pass-Criteria** | Ensure that the correct priority order of the message delivered to the device in the order HIGH MEDIUM and LOW. |
| **Comment** | None. |

## 7.3.8 Report Progress notes

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-16 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***progress-note*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.3.1 |
| **SCR Reference** | PAP-SEM-S-008 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***progress-note*** specifies the progress of the message within the PPG. There should be one progress note per stage of the process reported.<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with a status request to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response (***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>The PPG will then respond with a progress-note containing the following status details of the message.<br>1. The stage contains the text or code that indicates the stage of processing completed.<br>2. The Note contains textual description of the outcome of the stage completed<br>3. The Date/time stamp defines the stage completed.<br><br>The PPG will then handle the successful delivery if the Progress report is accepted. If the Progress Report is rejected or Not supported then an error code is returned as specified in the Appendix A Error Status Codes |
| **Pass-Criteria** | Ensure that the correct status is returned for the various states of the message delivery. |
| **Comment** | • |

## 7.3.9 Support capabilities entity in push message

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-17 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***ccq-message*** attribute and the PPG respond to the command appropriately Ensure that the Push capabilities are defined in the Push message |
| **Specification Reference** | [Push WAP247-PAP] Section 8.10 |
| **SCR Reference** | PAP-SEM-S-009 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***ccq-message*** requests the PPG to respond to the client capabilities for a specific device.<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with a ccq-message request to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response (***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>The PPG will then respond with a ***ccq-response*** containing the device capabilities of the requested device The query-ID can be used by the Push Initiator to associate with the related ccq-message...<br><br>Compose a Push message that has differing Push capabilities from those previously defined by including them in the Push message itself<br><br>The PPG will then handle the successful delivery of the request. If an error occurs then an error code is returned as specified in the Appendix A Error Status Codes |
| **Pass-Criteria** | Ensure that the correct client capabilities are returned upon the requested device... |
| **Comment** | None |

## 7.3.10 Return Status Code 3002

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-18 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately |
| **Specification Reference** | [Push WAP247-PAP] Section 8.12 |

| | |
|---|---|
| **SCR Reference** | PAP-SEM-S-010 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | PI that supports a PAP version after 1.0 should include the wap-pap-ver instruction in the submitted messages. Therefore allows the PPG to determine the versions the PI supports. If there is a mismatch then the PPG must report the various PAP versions the PPG supports. All PPG's must support PAP 1.0<br><br>The response ***badmessage-response*** sent from the PPG upon an invalid formatted message or that the protocol version is not supported by the PPG.<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code<br>If the message is unrecognisable then the Status code 2000 (Bad Request) is used<br>A fragment of the malformed message should be included in the bad message fragment attribute.<br><br>If the message is of a different version then the Status code 3002 (Version not supported) is used |
| **Test Procedure** | Submit a message that<br>1. has a malformed message – Which will result in Error code 2000<br>2. has incorrect version – Which will result in the error code 3002 |
| **Pass-Criteria** | If both Reponses are matching the correct Error codes as specified |
| **Comment** | None |

## 7.3.11 Detect the PAP version of a received message

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-19 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately. Detect the PAP version of the Initiator. |
| **Specification Reference** | [Push WAP247-PAP] Section 8.12 |
| **SCR Reference** | PAP-SEM-S-011 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PAP version number is placed in the public identifier [XML] of the DTD and will be changed with each revision of the version number. The filename of the DTD should also be changed so that it is easily identified with the public identifier<br>The PI will send a ***wap-pap-ver*** parameter which contains a ***supported versions*** parameter. The PPG will then accept by default version 1.0 of PAP protocol if no other versions are announced by the PI.<br><br>A PPG must include the ***wap-pap-ver*** processing instruction if the PI can support PAP versions above 1.0 otherwise it MUST NOT be included.<br><br>If the PPG does not support the PAP version used by the PI but a common version can be agreed the PPG must send a Status code 3002 (Version Unsupported) together with a bad message-response |
| **Test Procedure** | Submit a message that<br>has support for Multiple PAP Versions – Which will result in success not failure |
| **Pass-Criteria** | If both Reponses are matching the correct Acknowledgment No Error codes as specified |
| **Comment** | • |

## 7.3.12 Must send Versions supported processing instruction of PI > 1.0

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-20 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately. Detect the PAP version of the Initiator. |
| **Specification Reference** | [Push WAP247-PAP] Section 9.2 |
| **SCR Reference** | PAP-SEM-S-012 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PAP version number is placed in the public identifier [XML] of the DTD and will be changed with each revision of the version number. The filename of the DTD should also be changed so that it is easily identified with the public identifier<br>The PI will send a ***wap-pap-ver*** parameter which contains a ***supported versions*** parameter. The PPG will then accept by default version 1.0 of PAP protocol if no other versions are announced by the PI.<br><br>A PPG must include the ***wap-pap-ver*** processing instruction if the PI can support PAP versions above 1.0 otherwise it MUST NOT be included.<br><br>If the PPG does not support the PAP version used by the PI but a common version can be agreed the PPG must send a Status code 3002 (Version Unsupported) together with a bad message-response |
| **Test Procedure** | Submit a message that<br>has support for Multiple PAP Versions – Which will result in success not failure |
| **Pass-Criteria** | If both Reponses are matching the correct Acknowledgment No Error codes as specified |
| **Comment** | |

## 7.3.13 Report Supported versions

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-21 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately. Detect the PAP version of the Initiator. |

| | |
|---|---|
| **Specification Reference** | [Push WAP247-PAP] Section 9.2 |
| **SCR Reference** | PAP-SEM-S-013 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PAP version number is placed in the public identifier [XML] of the DTD and will be changed with each revision of the version number. The filename of the DTD should also be changed so that it is easily identified with the public identifier<br>The PI will send a ***wap-pap-ver*** parameter which contains a ***supported versions*** parameter. The PPG will then accept by default version 1.0 of PAP protocol if no other versions are announced by the PI.<br><br>A PPG must include the ***wap-pap-ver*** processing instruction if the PI can support PAP versions above 1.0 otherwise it MUST NOT be included.<br><br>If the PPG does not support the PAP version used by the PI but a common version can be agreed the PPG must send a Status code 3002 (Version Unsupported) together with a bad message-response<br>If a message has been sent by the PI that has been accepted by the PPG then the further message submitted must also sent with this version. |
| **Test Procedure** | Submit a message that<br>has support for Multiple PAP Versions – Which will result in success not failure |
| **Pass-Criteria** | If both Reponses are matching the correct Acknowledgment No Error codes as specified |
| **Comments** | |

## 7.3.14 Support sending Version 1.0

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-22 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately. Detect the PAP version of the Initiator. |
| **Specification Reference** | [Push WAP247-PAP] Section 9.3 |
| **SCR Reference** | PAP-SEM-S-014 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PAP version number is placed in the public identifier [XML] of the DTD and will be changed with each revision of the version number. The filename of the DTD should also be changed so that it is easily identified with the |

| | |
|---|---|
| | public identifier<br>The PI will send a ***wap-pap-ver*** parameter which contains a ***supported versions*** parameter. The PPG will then accept by default version 1.0 of PAP protocol if no other versions are announced by the PI.<br><br>A PPG must include the ***wap-pap-ver*** processing instruction if the PI can support PAP versions above 1.0 otherwise it MUST NOT be included.<br><br>If the PPG does not support the PAP version used by the PI but a common version can be agreed the PPG must send a Status code 3002 (***bad message - response*** Version Unsupported) together with a bad message-response<br><br>If a message has been sent by the PI that has been accepted by the PPG then the further message submitted must also sent with this version. |
| **Test Procedure** | Submit a message that<br>has support for PAP Version 1.0 – Which will result in success not failure |
| **Pass-Criteria** | If both Reponses are matching the correct Acknowledgment with No Error codes resulting. |
| **Comments** | |

## 7.3.15 Version Consistency

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-23 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG respond to the command appropriately. Check the PAP version of the Initiator. is consistent in its submissions |
| **Specification Reference** | [Push WAP247-PAP] Section 9.5 |
| **SCR Reference** | PAP-SEM-S-015 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PAP version number is placed in the public identifier [XML] of the DTD and will be changed with each revision of the version number. The filename of the DTD should also be changed so that it is easily identified with the public identifier<br>The PI will send a ***wap-pap-ver*** parameter which contains a ***supported versions*** parameter. The PPG will then accept by default version 1.0 of PAP protocol if no other versions are announced by the PI.<br><br>A PPG must include the ***wap-pap-ver*** processing instruction if the PI can support PAP versions above 1.0 otherwise it MUST NOT be included.<br><br>If the PPG does not support the PAP version used by the PI but a common version can be agreed the PPG must send a Status code 3002 (***bad message - response*** Version Unsupported) together with a bad message-response |

| | |
|---|---|
| | The PPG must send a message with the same version as previously submitted by the PI |
| **Test Procedure** | Send a message from the PI then resend the same message but with a different PI PAP version and an error should occur due to Inconsistency |
| **Pass-Criteria** | If both Reponses are matching the correct Acknowledgment with No Error codes resulting |
| **Comments** | |

### 7.3.16 Push Message Replacement

| | |
|---|---|
| **Test Case ID** | Push-PAP-2.2-int-24 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the **replace-push-id** attribute |
| **Specification Reference** | [Push WAP247-PAP] Section 8.2 |
| **SCR Reference** | PAP-SEM-S-016 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The presence of this **replace-push-id** parameter indicates to the PPG that the PI is requesting this Push message is replacing a previously submitted still pending push message id equal to **replace-push-id** in this push message<br><br>The absence of the **replace-push-id** attribute indicates that this push message MUST NOT replace any previously submitted push message i.e. it is a new submitted message. |
| **Test Procedure** | Send a message from the PI with a **Replace-Push-ID** matching the push ID of a previously sent message. Then the PPG will replace the content of the previous message with new content that is contained in this new message. |
| **Pass-Criteria** | The message received by the device is the latest message with the new content and not the old message content. |
| **Comments** | |

# 8. PPG Test Cases

There are 32 interoperability test cases for WAP 2.1 Push Proxy gateway Enabler.

## 8.1 Validation of Push Predicates

### 8.1.1 Validate confirmed Push is supported in the PPG

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-1 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that confirmed Push is supported by the PPG. |
| **Specification Reference** | [OMA-WAP-TS-PPGService] Section 5.1<br>(OTA-CO-S-002 OR OTA-CO-S-003) AND PPG-GEN-S-013 |
| **SCR Reference** | PPG-CO-S-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and receiptant Device MSISDN are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Confirmed push Message on the Push Initiator and send it to the PPG via a HTTP connection.<br>The PPG will respond with an Error Code defined in Appendix A depending upon the error condition during validation of the data contained in the Message<br>The PPG will then handle the successfully delivery of the message to the receiptant Device. |
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission. |
| **Comment** | |

## 8.2 Validation of Operations

## 8.2.1 Validation of Push Submission Rejection

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-2 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that an invalid pap push element is contained in the Push submission with respect to its Document Type Definition (DTD) by the initiator can be rejected by the PPG. |
| **Specification Reference** | Section 5.1.1 |
| **SCR Reference** | PPG-GEN-S-001 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and receiptant Device MSISDN are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator with an invalid pap push element and send it to the PPG via a HTTP connection.<br>The PPG will respond with a “Rejection” upon the invalidation of the data contained in the Message. |
| **Pass -Criteria** | The Initiator will accept the Rejection message from the PPG and an error will given to the user.<br>No Push will be sent OTA. |
| **Comment** | |

## 8.2.2 Validation of Transformed Messages

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-3 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that the PPG can transform the Push message entity in preparation for the Over the air delivery |
| **Specification Reference** | Section 5.1.1 |
| **SCR Reference** | PPG-GEN-S-002 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN's or addresses are pre configured on the PPG.<br><br>There are three types of addresses 1. The push proxy gateway Address, 2. The wireless Device address. 3. Result notification address<br><br>The Wireless Device address is the considered entity here and can have many formats i.e. an IP address or MSISDN / Subscriber number.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response (***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command (statusquery***-message***) can be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices.<br><br>The receiptant's address has to exist otherwise the message will not be sent as the user is unknown.<br><br>If the content being sent is not suitable for the Device's capabilities then Message Handling or Transformation can occur to format the content to be acceptable by the device. |
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission The content delivered to the device has been transformed by the PPG as the Device cold not support the original message format... |
| **Comments** | PPG/Initiator |

## 8.2.3 Validation of No Transform cache Control Directive

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-4 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that the PPG will NOT transform the Push message entity in preparation for the Over the air delivery as it has a NO transform cache control directive. |
| **Specification Reference** | Section 5.1.2.1.1 |
| **SCR Reference** | PPG-GEN-S-003 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>A PPG MUST NOT transform the body of any entity, which falls under the scope of a No-Transform cache control directive as defined in [RFC2616<br>Result notifications are returned by the PPG per receiptant if the result notification is requested the PI during the submission of a message.<br>If a cancel message is submitted the PPG may send back individual responses related to each receiptant's message or a response status for all the messages<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response (***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command (statusquery***-message***) should be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |
| **Pass-Criteria** | The Initiator will accept the ACKnowledgement from the PPG on a successful message submission The content delivered to the device has NOT been transformed by the PPG as defined by the NO transform cache control. |
| **Comment** | |

## 8.2.4 Validation of revising headers of transformed entities

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-5 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that the headers of all transformed entities must be revised as needed to correctly represent the transformed entity. |
| **Specification Reference** | Section 5.1.2.1.1 |
| **SCR Reference** | PPG-GEN-S-004 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>A PPG MUST transform the body of any entity, which falls under the scope of a Transform control directive as defined in [RFC2616<br><br>Result notifications are returned by the PPG per receiptant if the result notification is requested the PI during the submission of a message.<br><br>If a cancel message is submitted the PPG may send back individual responses related to each receiptant's message or a response status for all the messages<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator with the message type (**push-message)** and send it to multiple receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a Push Notification Response (***push-response***) depending upon the validation of the data contained in the Message.<br>Once a message has been submitted then a Status command ( ***statusquery-message***) should be sent to the PPG after the initial submission so allowing the PPG time to respond to query on each of the multiple pending pushes.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices.<br><br>The message headers are revised if the data has been transformed |
| **Pass-Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message when successful will then be placed on the Push Queue for Delivery |
| **Comment** | |

## 8.2.5 X-Wap-Application-ID header

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-6 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML Push Message containing the X-Wap-Application-Id header can be sent to the PPG and the PPG respond to the command appropriately. |
| **Specification Reference** | Section 5.1.2.1.2 |
| **SCR Reference** | PPG-GEN-S-005 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator with a correct and an Incorrect Application ID then send it to the PPG via a HTTP connection.<br>The PPG will then respond with a Result Notification Response (***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>If the header contains a [PushMsg] `absolute URI` format Application-ID for which an `app-assigned-code` has been registered with [OMNA], then the PPG MUST remove any [PushMsg] `app-assigned-code` format Application-ID (if present) from the header and then substitute this with the registered `app-assigned-code` format Application-ID for the `absolute URI` format Application-ID.<br><br>If the header contains a [PushMsg] `absolute URI` format Application-ID for which no `app-assigned-code` has been registered with [OMNA], the PPG MUST uses this value unless a [PushMsg] `app-assigned-code` format Application-ID is present. In this case (if the `app-assigned-code` format Application-ID is present), the `absolute URI` format Application-ID must be removed.<br><br>If no [PushMsg] `X-Wap-Application-Id` header is present in the push message, the PPG MUST, unless the client's default Application-ID is the WML user agent, add this header. If added, the Application-ID MUST is that of the WML user agent. |

| | |
|---|---|
| **Pass -Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message when successful will then be placed on the Push Queue for Delivery |
| **Comment** | |

## 8.2.6 X-Wap-Application-Id in numeric encoded format

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-7 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a valid XML Push Message containing the X-Wap-Application-Id header can be sent to the PPG and the PPG respond to the command appropriately and send the Numeric encoding format over the air to the device. |
| **Specification Reference** | Section 5.1.2.1.2 |
| **SCR Reference** | PPG-GEN-S-006 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator with a correct and an Incorrect Application ID then send it to the PPG via a HTTP connection.<br>The PPG will then respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>If the header contains a [PushMsg] `absolute URI` format Application-ID for which an `app-assigned-code` has been registered with [OMNA], then the PPG MUST remove any [PushMsg] `app-assigned-code` format Application-ID (if present) from the header and then substitute this with the registered `app-assigned-code` format Application-ID for the `absolute URI` format Application-ID.<br><br>If the header contains a [PushMsg] `absolute URI` format Application-ID for which no `app-assigned-code` has been registered with [OMNA], the PPG MUST uses this value unless a [PushMsg] `app-assigned-code` format Application-ID is present. In this case (if the `app-assigned-code` format Application-ID is present), the |

| | |
|---|---|
| | `absolute URI` format Application-ID must be removed.<br><br>If no [PushMsg] `X-Wap-Application-Id` header is present in the push message, the PPG MUST, unless the client's default Application-ID is the WML user agent, add this header. If added, the Application-ID MUST is that of the WML user agent. |
| **Pass -Criteria** | The Initiator will accept the Notification Response from the PPG on a successful or un-successful message submission. The message when successful will then be placed on the Push Queue for Delivery<br><br>The format of the X-Wap_Application_ID will be sent over the air by referencing the OMNA registered Numeric Encoded Value for the specific URI |
| **Comment** | |

## 8.2.7 Reportable message states

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-8 |
| **Test Object** | PPG/Initiator |
| **Test Case Test Case Description and Purpose** | Verify that a error message response and is sent when an invalid XML Push command is sent to the PPG and no message delivery is attempted |
| **Specification Reference** | Section 5.1.2.1.3 |
| **SCR Reference** | PPG-GEN-S-007 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test procedure** | Compose a Message on the Push Initiator and send it to one or more receipt ants on the PPG via a HTTP connection.<br>The PPG will respond with a message state depending upon the validation of the data contained in the Message.<br>FAILURE<br>The ***PAP-resultnotification*** can contain the following ***Message-state*** value "***undeliverable***" the Error ***code*** can convey a reason for failure "***transformation-failure***" If the message cannot be transformed properly.<br>SUCCESS |

| | |
|---|---|
| | If the message is successful then the ***PAP-resultnotification*** can contain the following ***Message-state*** value "***pending***" there is no Error ***code*** when the message is successful. |
| **Pass -Criteria** | The Initiator will accept the PAP_resultnotification Response from the PPG on a successful or un-successful message submission. The message will then be placed on the Push Queue for Delivery if successful. |
| **Comment** | |

## 8.2.8 Bearer Network Selection (QOS)

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-9 |
| **Test Object:** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG requiring a specific Quality of Service i.e. a defined Bearer and/or network to deliver the message. |
| **Specification Reference** | Section 5.1.2.2.2 |
| **SCR Reference** | PPG-GEN-S-008 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br>Push Initiator has been configured to the appropriate PPG server IP etc.<br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedure** | Compose a Message on the Push Initiator and send it to one or more receipt ants on the PPG via a HTTP connection.<br>The message will define various Bearers and/or Networks i.e. bearer:-GPRS,SMS Network:-GSM, IDEN,PDC<br>The PPG will respond with a message state depending upon the validation of the data contained in the Message.<br>FAILURE<br>The ***PAP-resultnotification*** can contain the following ***Message-state*** value "***undeliverable***" the Error ***code*** can convey a reason for failure "***An appropriate, implementation-dependant value***" The ***event-time*** contains "**Specific Time or estimated time of failure**" If the message cannot be delivered in the specific bearer and/or network. |

| | |
|---|---|
| **Pass-Criteria** | If the message is successful then the ***PAP-resultnotification*** can contain the following ***Message-state*** value ***"pending"***<br><br>The message method of delivery is as defined by the original push message and will be successfully delivered as defined by the network and bearer values.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |
| **Comment** | Note Multiple recipients is an Optional feature of PPG |

## 8.2.9 Reporting Session/registration Errors

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-10 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG requiring a specific Quality of Service i.e. a defined Bearer and/or network to deliver the message |
| **Specification Reference** | Section 5.1.2.2.3 |
| **Requirement:** | PPG-GEN-S-009 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Push Queue in PPG is empty<br><br>Push Initiator has been configured to the appropriate PPG server IP etc.<br><br>The Push content types and the receiptant Device MSISDN/s or address/es are pre configured on the PPG.<br><br>It is highly recommended to have a protocol analyzer to monitor traffic between the PPG and the Initiator. |
| **Test Procedures** | Compose a Message on the Push Initiator and send it to one or more receipt ants on the PPG via a HTTP connection.<br>The message will define various Bearers and/or Networks i.e. bearer:-GPRS,SMS Network:-GSM, IDEN,PDC<br>The PPG will respond with a message state depending upon the validation of the data contained in the Message.<br><br>In this case to verify the failure message response chooses a network/bearer that is not supported by the PPG. |
| **Pass-Criteria** | If the message cannot be delivered in the specific bearer and/or network.<br>It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***undeliverable***" the Error ***code*** can convey a reason for failure "***An appropriate, implementation-dependant value***" The ***event-time*** contains "**Specific Time or estimated time of failure**"<br><br>The message method of delivery is as defined by the original push message and will be successfully delivered as defined by the network and bearer values.<br><br>The PPG will then handle the successfully delivery of the message to the receiptant Device or Devices. |
| **Comment** | Note Multiple recipients is an Optional feature of PPG |

## 8.2.10 Delivery Time Constraints

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-11 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***deliver-after timestamp*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.1.2.2.4 |
| **SCR Reference** | PPG-GEN-S-010 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Prerequisites:** | The attribute Deliver after timestamp specifies the time and date by which the content must be delivered to the wireless device. Content that has aged prior to this date must NOT be transmitted. The date format must be in Co-ordinated Universal Time (UTC)<br>The PPG MUST reject the message submission from the initiator if the PPG does not support this function |
| **Test procedure:** | Compose a Message on the Push Initiator and send it with various Deliver after date and time stamps to a receiptant on the PPG via a HTTP connection.<br><br>If the message cannot be delivered in the specific time It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***expired***" the Error ***code*** can convey a reason for failure "***An appropriate, implementation-dependant value***" The ***event-time*** contains "**Specific Time or estimated time of failure**"<br><br>Create a message with a Deliver after date time stamp greater and less than present time<br>The PPG will then handle the successful delivery as the time stamp specifies<br>If the message is successful then the ***PAP-resultnotification*** can contain the following ***Message-state*** value ***"pending"*** |
| **Pass criteria:** | Ensure that the message that has a timestamp Less than present time is ignored and fails to deliver as the time has expired. The greater time stamp should deliver correctly after the expiry time has an elapsed in respect to current date and time as specified in the message attribute. |
| **Comment** | |

## 8.2.11 Delivery Method

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-12 |
| **Test object** | PPG/Initiator**.** |
| **Test case Description and Purpose** | Verify that a valid Confirmed / Unconfirmed OTA delivery method of the Push message is established by the PPG to the Device as defined by the submitted message. |
| **Specification Reference** | Section 5.1.2.2.5 |
| **SCR Reference** | PPG-GEN-S-011 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The delivery method attribute defines if the message is sent via a Confirmed or Unconfirmed Delivery.<br>The bearer can be either WSP or HTTP. If the OTA-HTTP is used and the PI indicates it accepts content from the client in the response to a confirmed push then the ***X-WAP-Push-Info*** header must contain the ***response*** attribute token |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with various Delivery methods i.e. Confirmed, Prefer confirmed, Confirmed-with-response, One-shot, Unconfirmed, Not specified via different bearers.<br><br>The PPG will respond to the PI on the confirmation of delivery to the device in the case of Confirmed Push (PO-Confirmed Push) |
| **Pass-Criteria** | Ensure that the message is delivered properly to the device and that the PI has had confirmation in the case of confirmed Push. |
| **Comment** | None |

## 8.2.12 Reported Unconfirmed Status

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-13 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***delivery method*** value set to UN Confirmed and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.1.2.2.5.1 |
| **Requirement:** | PPG-GEN-S-012 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |

| | |
|---|---|
| **Preconditions** | The delivery method attribute defines if the message is sent via a Un Confirmed Delivery.<br>The bearer can be either WSP or HTTP. If the OTA-HTTP is used and the PI indicates it accepts content from the client in the response to a un confirmed push then the ***X-WAP-Push-Info*** header must contain the ***response*** attribute token<br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with Unconfirmed delivery type to a receiptant on the PPG via a HTTP connection.<br>.<br>If the message is delivered in the specific time It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***delivered***" the **Delivery-method** conveys the method of delivery "***Unconfirmed***" The ***event-time*** contains "**Specific Time or estimated time of message delivery** |
| **Pass-Criteria** | Ensure that the correct Error Status code as defined in Appendix A is used in the response to incorrect or unsupported QOS. |
| **Comment** | None |

## 8.2.13 Reported Confirmed Status

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-14 |
| **Test Object** | PPG/Initiator**.** |
| **Test Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***delivery method*** value set to Confirmed and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.1.2.2.5.2 |
| **SCR Reference** | PPG-GEN-S-013 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The delivery method attribute defines if the message is sent via a Confirmed Delivery.<br>The bearer can be either WSP or HTTP. If the OTA-HTTP is used and the PI indicates it accepts content from the client in the response to a confirmed push then the ***X-WAP-Push-Info*** header must contain the ***response*** attribute token<br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with Confirmed delivery type to a receiptant on the PPG via a HTTP connection.<br><br>If the message is delivered in the specific time It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***delivered***" the **Delivery-method** conveys the method of delivery "***confirmed***" The ***event-time*** contains "**Specific Time or estimated time of message delivery** |
| **Pass-Criteria** | Ensure that the correct Error Status code as defined in Appendix A is used in the response to incorrect or unsupported QOS. |
| **Comment** | None. |

## 8.2.14 Result notification Message

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-15 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***resultnotification-message*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.2 |
| **SCR Reference** | PPG-GEN-S-014 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***resultnotification-message*** specifies the delivery status of the message. A result notification will be sent as soon as practical after the completion (successful or unsuccessful) of the Over the Air message delivery process<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test procedure** | Compose a Message on the Push Initiator and send it with the attribute ***resultnotification-message*** to a receiptant on the PPG via a HTTP connection. The PPG will respond with a Result Notification Response (***resultnotification-response***) depending upon the validation of the data contained in the Message. |
| **Pass-Criteria** | Ensure that the correct delivery status is returned upon the requested receipt ant |
| **Comment** | None. |

## 8.2.15 Pap Status Query

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-16 |
| **Test Object** | PPG/Initiator**.** |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***statusquery-message*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.3 |
| **SCR Reference** | PPG-GEN-S-015 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***statusquery-message*** specifies the progress of the message within the PPG. There should be one progress note per stage of the process reported.<br><br>The PPG MUST reject the message submission from the initiator if the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with a status request to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the Message.<br><br>The PPG will then reposnd with a progress-note containing the following status details of the message.<br>1. The stage contains the text or code that indicates the stage of processing completed.<br>2. The Note contains textual description of the outcome of the stage completed<br>3. The Date/time stamp defines the stage completed.<br><br>The PPG will then handle the successful delivery if the Progress report is accepted. If the Progress Report is rejected or Not supported then an error code is returned as specified in the Appendix A Error Status Codes |
| **Pass-Criteria** | Ensure that the correct status is returned for the various states of the message delivery. |
| **Comment** | • |

## 8.2.16 Delivery Cancellation

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-17 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***cancel-message*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.4 |
| **SCR Reference** | PPG-GEN-S-016 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute ***cancel-message*** requests the PPG to cancel the message delivery which it can perform when the message is within a state that can be cancelled.<br><br>The PPG MUST reject the message submission from the initiator if the the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with a ***cancel-message*** request to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(***resultnotification-response***) depending upon the validation of the data contained in the<br><br>In the case of Success<br>If the message cannot be delivered in the specific time It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***cancelled***" the Error ***code*** can convey a reason for failure "***An appropriate, implementation-dependant value***" The ***event-time*** conatins "**Specific Time or estimated time of failure** |
| **Pass-Criteria** | Ensure that the correct response is given upon the message cancellation. Also the message delivery does not occur on the device. |
| **Comment** | None |

## 8.2.17 Handling Message cancellation

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-18 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***cancel-message*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.4 |

| | |
|---|---|
| **SCR Reference** | PPG-GEN-S-017 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute cancel-message requests the PPG to cancel the message delivery which it can perform when the message is within a state that can be cancelled.<br><br>The PPG MUST reject the message submission from the initiator if the the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with a cancel-message request to a receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(resultnotification-response) depending upon the validation of the data contained in the Message it will be successfully or a failure.<br><br>If the PPG cannot assure cancellation of the message delivery it MUST reject the delivery cancellation<br>If the message cannot be cancelled in the specific time It results in a PAP-resultnotification which can contain the following Message-state value "cancellation not possible" the Error code 2008 can convey a reason for failure "The Push Id specified was found butcancellation is not possible" The event-time conatins "Specific Time or estimated time of failure |
| **Pass-Criteria** | Ensure that the cancellation not possible response is given upon the rejected message cancellation. Also the message delivery does not occur on the device. |
| **Comment** | None |

## 8.2.18 Validation of WSP specific transformation

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-19 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG encode the content in an implementation dependant manner suitable for delivery via WSP. |
| **Specification Reference** | Section 5.1.2.1.1.1 |
| **SCR Reference** | PPG-GEN-S-018 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PPG must support binary header encoding. It must also encode content entities into their compact binary format[WBXML] for transmission OTA-WSP unless it is known in advance that the receiptant device does not support the encoded type. |
| **Test Procedure** | Submit a message that will then be transformed into binary header encoding ready for transmission over OTA-WSP . Unless it is known in advance that the device does not support the encoded format |
| **Pass-Criteria** | The Device can receive the Push message successful in the encoded format. |
| **Comment** | • |

## 8.2.19 Validation of HTTP specific transformation

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-20 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG encode the content in an implementation dependant manner suitable for delivery via HTTP |
| **Specification Reference** | Section 5.1.2.1.1.1 |
| **SCR Reference** | PPG-GEN-S-019 |
| **Tool** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PPG must support binary header encoding. It must also encode content entities into their compact binary format[WBXML] or Deflate format for transmission OTA-HTTP unless it is known in advance that the receiptant device does not support the encoded type. |
| **Test Procedure** | Submit a message that will then be transformed into binary header encoding ready for transmission over OTA-HTTP . Unless it is known in advance that the device does not support the encoded format |
| **Pass-Criteria** | The Device can receive the Push message successful in either the WBXML or the Deflate encoded format. |
| **Comment** | |

## 8.2.20 Validation for Push message Replacement

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-21 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG containing the ***replace-message*** attribute and the PPG respond to the command appropriately |
| **Specification Reference** | Section 5.1.1.1 |
| **SCR Reference** | PPG-GEN-S-020 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The attribute replace-message requests the PPG to replace a previously posted message with a new one before it is delivered. The PPG can only perform this replacement when the message is within a state that can be processed ie it is in the |

| | |
|---|---|
| | Queue.<br><br>The PPG MUST reject the message submission from the initiator if the the PPG does not support or honour this function with the appropriate error code |
| **Test Procedure** | Compose two Messages on the Push Initiator and send one of the messages with a replace-message request together with the previously submitted message ID of the other receiptant on the PPG via a HTTP connection.<br>The PPG will respond with a Result Notification Response(resultnotification-response) depending upon the validation of the data contained in the<br><br>In the case of Success<br>If the message cannot be delivered in the specific time It results in a PAP-resultnotification which can contain the following Message-state value "replaced" the Error code can convey a reason for failure "An appropriate, implementation-dependant value" The event-time conatins "Specific Time or estimated time of failure |
| **Pass-Criteria** | Ensure that the correct response is given upon the message replacement. Also the message delivery occurs with the newer replacement message on the device. |
| **Comments** | |

## 8.2.21 Validation of binary header encoding

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-22 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG encode the headers in an implementation dependant manner suitable for delivery via WSP. |
| **Specification Reference** | Section 5.1.2.1.1.1 |
| **SCR Reference** | PPG-GEN-S-021 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PPG must support binary header encoding. It must encode the headers into their binary format for transmission viaOTA-WSP/HTTP unless it is known in advance that the receiptant device does not support the encoded type. |
| **Test Procedure** | Submit a message that will then be transformed into binary header encoding ready for transmission over OTA-WSP . Unless it is known in advance that the device does not support the encoded format |
| **Pass-Criteria** | The Device can receive the Push message successfully in the encoded format. |
| **Comments** | |

## 8.2.22 Validation of content encoding using WBXML

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-23 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG encode the content in an implementation dependant manner suitable for delivery via WSP/HTTP. |
| **Specification Reference** | Section 5.1.2.1.1.1 |
| **SCR Reference** | PPG-GEN-S-022 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PPG must support binary header encoding. It must encode the content into their compact binary format [WBXML]for transmission via OTA-WSP/HTTP unless it is known in advance that the receiptant device does not support the encoded type. |
| **Test Procedure** | Submit a message that will then have the content transformed into compact binary format ready for transmission over OTA-WSP/HTTP . Unless it is known in advance that the device does not support the encoded format |
| **Pass-Criteria** | The Device can receive the Push message successfully in the encoded format. |
| **Comments** | |

## 8.2.23 Validation of content encoding using deflate

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-24 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML Push message can be sent to the PPG and the PPG encode the content in an implementation dependant manner suitable for delivery via HTTP. |
| **Specification Reference** | Section 5.1.2.1.1.1 |
| **SCR Reference** | PPG-GEN-S-023 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The PPG must support deflate encoding. It must encode the content into their compact binary format [WBXML] or Deflatefor transmission via OTA- |

| | |
|---|---|
| | WSP/HTTP unless it is known in advance that the receiptant device does not support the encoded type. |
| **Test Procedure** | Submit a message that will then have the content transformed into compact binary format or Deflate ready for transmission over OTA-WSP/HTTP . Unless it is known in advance that the device does not support the encoded format |
| **Pass-Criteria** | The Device can receive the Push message successfully in the encoded format. |
| **Comments** | |

## 8.2.24 Validation of Delivery method Confirmed with response

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-25 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid Confirmed with response OTA delivery method of the Push message is established by the PPG to the Device as defined by the submitted message. |
| **Specification Reference** | Section 5.1.1.2 |
| **SCR Reference** | PPG-GEN-S-024 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The delivery method attribute defines if the message is sent via a Confirmed with response Delivery.<br>The bearer is HTTP. If the OTA-HTTP is used and the PI indicates it accepts content from the client in the response to a confirmed push then the ***X-WAP-Push-Info*** header must contain the ***response*** attribute token |
| **Test Procedure** | Compose a Message on the Push Initiator and send it with various Delivery methods ie Confirmed , Preferconfirmed, Confirmed-with-response, Oneshoot, Unconfirmed, Notspecified via different bearers.<br><br>The PPG will respond to the PI on the confirmation of delivery to the device in the case of Confirmed Push with the response the client accepts content from the PI<br><br>The PPG MUST reject the message submission from the initiator if the the PPG does not support or honour this function with the appropriate error code |
| **Pass-Criteria** | Ensure that the message is delivered properly to the device and that the PI has had confirmation of content in the case of confirmed Push. |
| **Comments** | |

## 8.2.25 Validation of selection of Push OTA Protocol

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-26 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that the PPG can select the OTA protocol [WSP /HTTP]for connection orientated push in an implementation dependant manner. |
| **Specification Reference** | Section 5.1.2.2.1 |
| **SCR Reference** | PPG-GEN-S-025 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | |
| **Test Procedure** | Compose a Message on the Push Initiator and send it via connection oriented.<br>When the PPG receives the message it may send it via PO or TO depending on the setup and configuration of the device and network.<br>If it is PO-TCP the PPG will then send it via HTTP. If the PPG sends it via TO-TCP then the decision is made by sending a Session Initiated request to the terminal. Then the terminal will decide which contact point OTA WSP or OTA HTTP to use to establish a connection. |
| **Pass-Criteria** | The message is successfully received by the device via any of the OTA protocols. |
| **Comments** | |

## 8.2.26 Validation of result-notification-message

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-27 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that the PI can support resultnotification and the PI can accept content from the client in its response to a confirmed push |
| **Specification Reference** | Section 5.2.2 |
| **SCR Reference** | PPG-GEN-S-026 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The Pap resultnotification message indicates the reportable message status which includes the message state and other information. The status should reflect the message just before the limits of practically sending the result notification.<br>Assuming the PI can accept content from the client in its response to a confirmed push. If the PI does not indicate that it accepts content from the client in response, the content entity must not be present when the |

| | |
|---|---|
| | resultnotification message is returned to the PI. |
| **Test Procedure** | Send a confirmed message from the PI with a content request in which the PPG will return the content acceptable by the device. |
| **Pass-Criteria** | The content that is passed to the PI matches what the device sends via its resultnotification. |
| **Comments** | |

### 8.2.27 Validation of Oneshot delivery status

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-28 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid XML One Shot Push message can be sent to the PPG and the PPG onlt attempt delivery once. |
| **Specification Reference** | Section 5.1.2.2.5.3 |
| **SCR Reference** | PPG-GEN-S-027 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | If the message can be delivered successfully. It results in a ***PAP-resultnotification*** which can contain the following ***Message-state*** value "***delivered***" then the Delivery method "***One Shoot***" The ***event-time*** conatins "**Specific Time or estimated time of failure** |
| **Test Procedure** | Send a message from the PI as a Oneshot type. The PPG will then attempt message delivery only once. and the response |
| **Pass-Criteria** | The message is successfully received by the device.. |
| **Comments** | |

## 8.3 Validation of Client Addressing

### 8.3.1 Validation of Client Addressing

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-29 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a valid Client Addressing methods are used when initating a message to the PPG |

| | |
|---|---|
| **Specification Reference** | Section 6 |
| **SCR Reference** | PPG-ADD-S-001 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | Create a message with the clients address in either a special textual address format or as a network-specific address. The PPG will have the responsibility of transforming either format into the match format response.<br>The PPG will respond with the address value in the same format as the submission |
| **Test Procedure** | Send a message from the PI with a client address that has a matching address from the PPG in the response. The PPG will transform the address format suitable for the deliver over the wireless network.. |
| **Pass-Criteria** | The message received by the device is accepted and the PI has accepted the format of the response from the PPG for the submitted message. |
| **Comments** | |

## 8.3.2 Validation of User defined identites

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-30 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a user defined identity can be sent to the PPG |
| **Specification Reference** | Section 6 |
| **SCR Reference** | PPG-ADD-S-002 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The presence of a user-defined identity are arbitary values that are mapped to wireless network addresses in an unspecific manner. The PPG has control over which bearer address will be used. A client can have one or more bearer identities |
| **Test Procedure** | Send a message from the PI with a User defined identity. The PPG will accept this message for processing and deliver the successful message via the bearer as choosen by PPG to the client. |
| **Pass-Criteria** | The message is successfully received by the device and the PI has accepted the format of the response from the PPG for the submitted message. . |
| **Comments** | |

## 8.3.3 Validation of Device Addresses

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-31 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a device Address can be sent to the PPG |
| **Specification Reference** | Section 6 |
| **SCR Reference** | PPG-ADD-S-003 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |
| **Preconditions** | The presence of a device address from well known wireless addresses spaces<br><br>The PPG has control over which bearer address will be used. A client can have one or more bearer identities. The bearer level address may invoke a point to multipoint delivery in the wireless network ie cell broadcast. In this case there must be a single result notification if one has been requested. |
| **Test Procedure** | Send a message from the PI with a device address. The PPG will accept this message for processing and deliver the successful message via the bearer as choosen by PPG to the client. |
| **Pass-Criteria** | The message is successfully received by the device and the PI has accepted the format of the response from the PPG for the submitted message. . |
| **Comments** | |

## 8.3.4 Validation of Client Address format

| | |
|---|---|
| **Test Case ID** | Push-PPG-2.2-int-32 |
| **Test Object** | PPG/Initiator |
| **Test Case Description and Purpose** | Verify that a Client Address can be sent to the PPG and be successfully accepted if supported |
| **Specification Reference** | Section 6 |
| **SCR Reference** | PPG-ADD-S-004 |
| **Tools** | PUSH INITIATOR |
| **Test Code/Files** | NON APPLICABLE |

| | |
|---|---|
| **Preconditions** | The external representation of addresses processed by the PPG is defined using ABNF [RFC2234]. The PPG will parse the address in this format and PPG will determine if it supports the specified address type or not.<br><br>The PPG has control over which bearer address will be used. A client can have one or more bearer identities.<br><br>MSISDN -/TYPE=PLMN<br>`WAPPUSH=+155519990730/TYPE=PLMN@ppg.carrier.com`<br>`; device address for a phone number of some wireless network`<br><br>USER …../TYPE=USER<br>`WAPPUSH=john.doe%40wapforum.org/TYPE=USER@ppg.carrier.com`<br>`; user-defined identifier for john.doe@wapforum.org`<br><br>`wappush=47397547589/type=user@carrier.com`<br>`; user-defined identifier for 47397547589`<br><br>Ipv4 -- /TYPE=IPV4<br>`WAPPUSH=195.153.199.30/TYPE=IPv4@ppg.carrier.com`<br>`; device address for an IP v4 address`<br>Ipv6 /TYPE=IPV6<br>`WAPPUSH=FEDC:BA98:7654:3210:FEDC:BA98:7654:3210/TYPE=IPv6@carrier.com`<br>`; device address for an IP v6 address`<br><br>Man -/TYPE=MAN<br>Escape Value -/TYPE=<br>WAPPUSH=12345678/TYPE=MAN@ppg.carrier.com<br>; device address for a MAN addressStatic Conformance Requirements (Normative |
| **Test Procedure** | Send a message from the PI with a client address. The PPG will accept this message for processing and deliver the successful message if supported via the bearer as choosen by PPG to the client. |
| **Pass-Criteria** | The message is successfully received by the device and the PI has accepted the format of the response from the PPG for the submitted message. . |
| **Comments** | |

# Appendix A. Error Status Codes

The status code is a 4 digit value. The first gigit of the ststus code indicates the class of the code. There are 5 classes

1XXX: Success

2XXX- Client Error

3XXX- Server Error

4XXX- Service Failure

5XXX- Mobile device Abort

| | | | Response Rsult | cancel-result | resultnotification-message | resultnotification-response | statusquery-result | ccq-response | badmessage-response |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Description** | | | | | | | | |
| 1000 | OK | The request succeeded. | | x | x | x | x | x | |
| 1001 | Accepted for Processing | The request has been accepted for processing. | x | | | | | | |
| 2000 | Bad Request | Not understood due to malformed syntax. | x | x | | x | x | x | x |
| 2001 | Forbidden | The request was refused. | x | x | x | | x | x | |
| 2002 | Address Error | The client specified was not recognised. | x | x | x | | x | x | |
| 2003 | Address Not Found | The address specified was not found. | | x | | | x | | |
| 2004 | Push ID Not Found | The Push ID specified was not found. | x | x | | | x | | |
| 2005 | Capabilities Mismatch | The capabilities assumed by the PI were not acceptable for the client specified. | x | | x | | | | |
| 2006 | Required Capabilities Not Supported | The input is in a form not supported by the client. | x | | x | | | | |
| 2007 | Duplicate Push ID | The Push ID supplied is not unique | x | | | | | | |

| | | | Response Rsult | cancel-result | resultnotification-message | resultnotification-response | statusquery-result | ccq-response | badmessage-response |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Description** | | | | | | | | |
| | | within the PPG. | | | | | | | |
| 2008 | Cancellation not possible | The Push ID specified was found, but cancellation is not possible | x | x | x | | | | |
| 3000 | Internal Server Error | Server could not fulfil request due to internal error. | x | x | | | x | x | |
| 3001 | Not Implemented | Server does not support the requested operation. | | x | | | x | x | |
| 3002 | Version not Supported | The server refuses to support the protocol version indicated. | | | | | | | x |
| 3003 | Not Possible | Action not possible because message is no longer available. | | x | | | x | | |
| 3004 | Capability Matching not Supported | The PPG does not support client capability information provided in a push message. | x | | | | | | |
| 3005 | Multiple Addresses Not Supported | The PPG does not support an operation that specified multiple recipients. | x | x | | | x | | |
| 3006 | Transformation Failure | The PPG was unable to perform a transformation on the message. | x | | x | | x | | |
| 3007 | Specified Delivery Method Not Possible | The PPG could not perform the confirmed or unconfirmed delivery specified. | x | | x | | x | | |
| 3008 | Capabilities Not Available | Client capabilities for the specified client are not available. | | | | | | x | |
| 3009 | Required Network Not Available | The network requested is not available. | x | | x | | x | | |
| 3010 | Required Bearer Not Available | The bearer requested is not available. | x | | x | | x | | |
| 3011 | Replacement Not Supported | The PPG does not support the replace operation | x | | | | | | |
| 3012 | One-shot Not Supported | The PPG or the bearer does not | x | | | | | | |

| | | | Response Rsult | cancel-result | resultnotification-message | resultnotification-response | statusquery-result | ccq-response | badmessage-response |
|---|---|---|---|---|---|---|---|---|---|
| **Code** | **Description** | | | | | | | | |
| | | support one-shot delivery. | | | | | | | |
| 4000 | Service Failure | The service failed. The client may re-attempt the operation. | | | x | | x | | |
| 4001 | Service Unavailable | The server is busy. | | | x | | x | | |
| 5xxx | Mobile Client Aborted | The mobile client aborted the operation. | | | x | | x | | |

# Appendix B. Change History (Informative)

## B.1 Approved Version History

| Reference | Date | Description |
|---|---|---|
| n/a | n/a | No prior version –or- No previous version within OMA |

## B.2 Draft/Candidate Version 2.2 History

| Document Identifier | Date | Section | Description |
|---|---|---|---|
| Draft Versions<br>OMA-ETS-Push-V2_2 | 03 Nov 2006 | 7 | Updates to the OTA section to include the 2.2 revision For security and SMS concatenation. |
| | 20 Nov 2006 | All | Updates after review with IOP-BRO |
| | 22 July 2007 | 5.0 | Update to include Provisioning server |
| | 18 Oct 2007 | n/a | IOP WG agreed, ref # OMA-IOP-2007-0210-INP_Push_2.2_ETS |
| Candidate Version<br>OMA-ETS-Push-V2_2 | 06 Nov 2007 | n/a | Status changed to candidate. TP R&A 2007-10-24 to 2007-11-06, TP doc. ref # OMA-TP-2007-0441-INP_ETS_Push_V2_2_for_ candidate_approval |