SIP-DECT XML Terminal Interface for Mitel 600 DECT Phone Family
# TABLE OF CONTENTS

1  **INTRODUCTION** ........................................................................................................... 2  
   1.1  **PURPOSE** ................................................................................................................ 2  
   1.2  **VERSION CHANGES** .............................................................................................. 2  
2  **OVERVIEW** .................................................................................................................. 3  
   2.1  **CONCEPT** ................................................................................................................ 3  
      2.1.1  **XML BROWSER** ............................................................................................ 3  
      2.1.2  **XML-APPLICATIONS** .................................................................................. 3  
         2.1.2.1  External Call Lists .......................................................................................... 3  
         2.1.2.2  XML User Presence ...................................................................................... 4  
         2.1.2.3  XML Server Application Menu .................................................................... 4  
         2.1.2.4  Optional Configurable External XML Applications ...................................... 4  
         2.1.2.5  XML PP State Notification .......................................................................... 4  
         2.1.2.6  XML PP FAC Notification .......................................................................... 6  
         2.1.2.7  XML PP Call Completion Notification .......................................................... 6  
         2.1.2.8  XML PP Call Park ......................................................................................... 6  
         2.1.2.9  XML PP Call Unpark ...................................................................................... 6  
         2.1.2.10 XML PP Call Pickup ..................................................................................... 6  
         2.1.2.11 XML PP Call Take ......................................................................................... 6  
         2.1.2.12 XML PP Call Forward ................................................................................... 6  
         2.1.2.13 XML PP Call Routing .................................................................................. 6  
         2.1.2.14 XML PP Call Protection .............................................................................. 6  
         2.1.2.15 XML PP Voice Box ...................................................................................... 6  
         2.1.2.16 UTF-8 Support .............................................................................................. 7  
   2.1.3  **XML USING HTTP** .......................................................................................... 7  
   2.1.4  **XML CLIENT BASED INTERFACE** .................................................................. 8  
   2.1.5  **PROTOCOL ARCHITECTURE** ......................................................................... 9  
3  **DETAILS** .................................................................................................................... 10  
   3.1  **PP AND USER INTERFACE REQUIREMENTS** .................................................... 10  
      3.1.1  **EXTERNAL CALL LISTS** ............................................................................... 11  
      3.1.2  **XML USER PRESENCE** ............................................................................... 12  
      3.1.3  **XML SERVER APPLICATION MENU** ........................................................... 13  
      3.1.4  **OPTIONAL CONFIGURABLE EXTERNAL XML APPLICATIONS** ............ 13  
      3.1.5  **XML PP STATE NOTIFICATION** ................................................................ 13  
      3.1.6  **XML PP FAC NOTIFICATION** ................................................................... 13  
      3.1.7  **XML CALL COMPLETION** ......................................................................... 13  
      3.1.8  **XML CALL PARK HOOK** .......................................................................... 13  
      3.1.9  **XML CALL UNPARK HOOK** ..................................................................... 14
3.1.10 XML CALL PICKUP HOOK .......................................................... 14
3.1.11 XML CALL TAKE HOOK .......................................................... 14
3.1.12 XML CALL FORWARD HOOK ..................................................... 14
3.1.13 XML CALL ROUTING HOOK ...................................................... 14
3.1.14 XML CALL PROTECTION HOOK ............................................... 15
3.1.15 XML VOICE BOX HOOK .......................................................... 15
3.2 OMM REQUIREMENTS ..................................................................... 15
  3.2.1 EXTERNAL CALL LISTS ............................................................. 17
  3.2.2 XML USER PRESENCE ............................................................... 18
  3.2.3 XML SERVER APPLICATION MENU ........................................ 18
  3.2.4 OPTIONAL CONFIGURABLE EXTERNAL XML APPLICATIONS ...... 18
  3.2.5 XML PP STATE NOTIFICATION .................................................. 19
  3.2.6 XML PP FAC NOTIFICATION ................................................... 19
  3.2.7 XML PP CALL COMPLETION NOTIFICATION .............................. 19
  3.2.8 XML PP CALL PARK ................................................................. 20
  3.2.9 XML PP CALL UNPARK ............................................................ 20
  3.2.10 XML PP CALL TAKE .............................................................. 20
  3.2.11 XML PP CALL FORWARD ....................................................... 21
  3.2.12 XML PP CALL ROUTING ........................................................ 21
  3.2.13 XML PP CALL PROTECTION .................................................... 21
  3.2.14 XML PP VOICE BOX ............................................................... 22
3.3 PBX Or XML SERVER REQUIREMENTS ........................................ 22
  3.3.1 EXTERNAL CALL LISTS ............................................................ 22
  3.3.2 XML USER PRESENCE ............................................................. 23
  3.3.3 XML SERVER APPLICATION MENU ........................................ 23
  3.3.4 OPTIONAL CONFIGURABLE EXTERNAL XML APPLICATIONS ...... 23
  3.3.5 XML PP STATE NOTIFICATION .................................................. 24
  3.3.6 XML FAC NOTIFICATION ......................................................... 24
  3.3.7 XML CALL COMPLETION NOTIFICATION .................................. 24
  3.3.8 XML CALL PARK ................................................................. 24
  3.3.9 XML CALL UNPARK ............................................................. 24
  3.3.10 XML CALL PICKUP ............................................................... 24
  3.3.11 XML CALL TAKE ................................................................. 24
  3.3.12 XML CALL FORWARD ............................................................ 24
  3.3.13 XML CALL ROUTING ............................................................ 25
  3.3.14 XML CALL PROTECTION ........................................................ 25
3.3.15 XML VOICE BOX .................................................................................................................. 25

3.4 HTTP PROTOCOL .................................................................................................................. 25

3.4.1 HTTP GET REQUEST HEADER ........................................................................................ 25

3.4.2 URI REQUESTS .................................................................................................................. 26

3.4.2.1 External Call Lists .......................................................................................................... 26

3.4.2.2 XML User Presence ...................................................................................................... 28

3.4.2.3 XML Server Application Menu ...................................................................................... 28

3.4.2.4 Optional Configurable External XML Applications ...................................................... 29

3.4.2.5 XML PP State Notification ............................................................................................ 29

3.4.2.6 XML PP FAC Notification ........................................................................................... 30

3.4.2.7 XML PP Call Completion Notification ......................................................................... 31

3.4.2.8 XML PP Call Control Hooks ......................................................................................... 31

3.4.3 XML-DOCUMENTS ............................................................................................................. 32

3.4.3.1 “TextMenu” XML Object Usage .................................................................................... 32

3.4.3.3.1 “Get Call List” XML – Response .............................................................................. 35

3.4.3.1.2 “Delete Call List” (Entry) XML – Response ............................................................... 36

3.4.3.1.3 “Exit Call Lists” XML – Response ............................................................................ 38

3.4.3.2 “TextScreen” XML Object Usage ................................................................................ 38

3.4.3.2.1 XML “TextScreen” Response / Got As Call List Details ............................................ 40

3.4.3.3 “PhoneStatus” XML Object Usage ............................................................................. 41

3.4.3.4 “InputScreen (single)” XML Object Usage .................................................................... 43

3.4.3.5 “InputScreen (multiple)” XML Object Usage ................................................................ 45

3.4.3.6 “PhoneConfiguration” XML Object Usage ................................................................. 45

3.4.3.7 “PhoneExecute” XML Object Usage ........................................................................... 47

3.4.4 SIP NOTIFY EVENTS ......................................................................................................... 48

3.5 XML APPLICATION CONFIGURATION VIA OMP ................................................................ 50

3.5.1 BUILT-IN XML APPLICATIONS .................................................................................... 51

3.5.2 ADDITIONAL USER DEFINED XML APPLICATIONS ................................................... 52

4 APPENDIX ................................................................................................................................. 54

4.1 USEABLE ICONS ................................................................................................................... 54

4.2 DETAILED EXAMPLES ......................................................................................................... 57

4.2.1 DIAL FROM CALLER LIST ............................................................................................... 57

4.2.2 DELETE COMPLETE REDIAL LIST ............................................................................... 58

4.2.3 DELETE A CALLER LIST ENTRY ....................................................................................... 61

5 RESTRICTIONS ........................................................................................................................ 66

5.1 XML CONCERNED RESTRICTIONS .................................................................................... 66

5.2 DIRECT DIAL RESTRICTIONS ............................................................................................. 67

5.3 UPGRADING A5000 CALL LISTS TO OMM SIP-DECT 4.0 ............................................. 67
## Revision Control Information

<table>
<thead>
<tr>
<th>Date</th>
<th>Author</th>
<th>Version</th>
<th>Changes</th>
</tr>
</thead>
</table>

## Referenced Documents

<table>
<thead>
<tr>
<th>Reference</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>/1/ aad-0384</td>
<td>SIP-DECT OM Application XML Interface (Rel. 6.0)</td>
</tr>
<tr>
<td>/2/ PA-001008-05-00</td>
<td>Development Guide XML API for Aastra SIP Phones</td>
</tr>
<tr>
<td>/3/</td>
<td>Technical Information SIP-DECT 4.0 XML handset provisioning by Julian Zelina / Aastra Deutschland GmbH, Berlin</td>
</tr>
<tr>
<td>/4/</td>
<td>Technical Information MX-ONE: Link SIP-DECT 4.0 with MX-ONE by Julian Zelina / Aastra Deutschland GmbH, Berlin</td>
</tr>
</tbody>
</table>

## Definitions and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DECT</td>
<td>Digital Enhanced Cordless Telecommunications</td>
</tr>
<tr>
<td>FAC</td>
<td>Feature Access Code</td>
</tr>
<tr>
<td>HTTP</td>
<td>HyperText Transport Protocol</td>
</tr>
<tr>
<td>IP</td>
<td>Internet Protocol</td>
</tr>
<tr>
<td>MSC</td>
<td>Message Sequence Chart</td>
</tr>
<tr>
<td>OMMAXI</td>
<td>OpenMobility Application XML Interface</td>
</tr>
<tr>
<td>OM AVM</td>
<td>OpenMobility Application XML Interface (Rel. 6.0)</td>
</tr>
<tr>
<td>OMM</td>
<td>OpenMobility Manager</td>
</tr>
<tr>
<td>OMP</td>
<td>OpenMobility Manager Portal</td>
</tr>
<tr>
<td>PP</td>
<td>Portable Part, a DECT phone</td>
</tr>
<tr>
<td>RFP</td>
<td>Radio Fixed Part, a DECT base station</td>
</tr>
<tr>
<td>SIP</td>
<td>Session Initiation Protocol</td>
</tr>
<tr>
<td>SSL</td>
<td>Security Socket Layer - Transport Layer Security</td>
</tr>
<tr>
<td>TCP</td>
<td>Transport Control Protocol</td>
</tr>
<tr>
<td>URI</td>
<td>Uniform Resource Locator</td>
</tr>
<tr>
<td>URL</td>
<td>Uniform Resource Identifier</td>
</tr>
<tr>
<td>XML</td>
<td>Extensible Markup Language</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Purpose

This document describes the realization of external applications used within a Mitel SIP-DECT system.

- It describes the consequences on the PP.
- It describes the interface and protocol to the external application.
- It describes features, functions and restrictions in conjunction with the external application.

The document describes the following applications used with Mitel SIP-DECT Phones and the Mitel SIP-DECT system.

- Call list handling in an external PBX
- XML User Presence
- XML Server Application menu
- Optional configurable external XML applications
- XML PP State notification

Only the Mitel SIP DECT Phones support XML applications.

Note: See the restrictions for XML applications in chapter 5.

1.2 Version Changes

0.1: Initial draft version
0.2 - 0.7: Draft versions
>= 0.8: Valid versions
<= 0.24: SIP-DECT Version < 4.0
0.25: SIP-DECT Version 4.0
0.26: SIP-DECT Version 5.0
> (0.28) 6.0(.x): Mitel SIP-DECT Version >= 6.0
2 Overview

2.1 Concept

The OMM release 2.0 introduces first an XML interface to manage the OMM by external applications. As known from Mitel SIP phones ([2]) an XML interface is even useful to manage devices especially for additional applications. So the OMMs XML interface is extended in the Mitel SIP-DECT system to manage also applications for the PPs. The “Development Guide XML API for Mitel SIP Phones” ([2]) is the base for this realization.

The goal for OMM SIP-DECT release 3.0 is an XML browser based on this development guide to set up different applications. The external call lists handling is the first XML application step using this new XML browser.

2.1.1 XML Browser

In the first step of the XML browser handles those Mitel IP Phone XML objects which are necessary for the external call lists handling. These are mainly text based XML objects ([2]) to be displayed at the PP as follows:

- **TextMenu** XML object
- **TextScreen** XML object
- **PhoneStatus** XML object

Further the XML browser handles additional IP Phone XML objects which are necessary for further XML applications. Additionally the following XML objects ([2]) are supported by the OMM and the PPs:

- **InputScreen** XML object
- **PhoneConfiguration** XML object
- **PhoneExecute** XML object

2.1.2 XML-Applications

2.1.2.1 External Call Lists

The first application to be introduced in the OMM is the handling of the call lists

- Redial list (list of all outgoing calls)
- Caller list (list of all incoming calls)

which are part of the PBX and be managed there. Optionally the PPs can use the call lists of the PBX instead of manage them locally. This is done by a local setting in the PP.
Each time a PP user enters an external call list menu a DECT air connection to the OMM is established to retrieve the list from the PBX.

2.1.2.2 XML User Presence

To use the user presence supported by an XML server application for a Mitel DECT Phone its option menu has to be extended by a “Presence” entry. This is done by an OMP configuration for the related “XML Presence” application (see 3.5).

Each time a PP user enters the “Presence” menu entry a DECT air connection to the OMM is established to access the servers “XML Presence application”.

2.1.2.3 XML Server Application Menu

To use the server application menu supported by an XML server application for a Mitel DECT Phone its OMM menu (contained in the system menu) has to be extended by a “Server menu” entry. This is done by an OMP configuration for the related “XML Server menu” application (see 3.5).

Each time a PP user enters the “Server menu” entry a DECT air connection to the OMM is established to access the servers “XML Server application menu”.

2.1.2.4 Optional Configurable External XML Applications

To use further optional XML server applications by an external server for a Mitel DECT Phone its option menu has to be extended by an “Applications” entry. This is done by an OMP configuration for at least one XML application creation (see 3.5). Up to ten XML applications can be created.

Each time a PP user enters the “Applications” entry a DECT air connection to the OMM is established to select one of a servers “XML application”.

Kind of applications might be:

- Hospital applications
- Directory
- Call center
- ...

2.1.2.5 XML PP State Notification

To notify PP/handset states to external server an OMP configuration for the “Event actions” application (see 3.5) has to be done.

The OMM supports the usage of action URIs notifications as defined in [2] for PP call states:
- End of the boot sequence  
  action uri startup  
  /* indicates a location registration of the PP */

- Successful registration  
  action uri registered  
  /* indicates a SIP registration of the PP */

- On-hook  
  action uri onhook

- Off-hook  
  action uri offhook

- Incoming call  
  action uri incoming

- Outgoing call  
  action uri outgoing

- Time based  
  action uri poll  
  /* Not supported by the OMM */

- SIP notify  
  action uri sip notify

- Connect  
  action uri connected

- Disconnect  
  action uri disconnected

- Registration event  
  action uri registration event  
  /* indicates a SIP registration attempt of the PP */

Additionally supported by the OMM for PP states:

- Deregistration  
  action uri deregistered  
  /* indicates a SIP deregistration of the PP */

- Silent charging on  
  action uri silent charging on

- Silent charging off  
  action uri silent charging off

- Detach  
  action uri detach

The following table shows an extract for possible states that can be mapped from action URIs notifications:

<table>
<thead>
<tr>
<th></th>
<th>pa1</th>
<th>pa2</th>
<th>pa3</th>
<th>pa4</th>
<th>pa5</th>
<th>pa6</th>
<th>pa7</th>
<th>pa8</th>
<th>pa9</th>
<th>pa10</th>
<th>pa11</th>
<th>pa12</th>
<th>pa13</th>
<th>pa14</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>{sichao}</td>
<td>{sichao}</td>
<td>{boot}</td>
<td>{reg}</td>
<td>{dereg}</td>
<td>{onho}</td>
<td>{offho}</td>
<td>{in}</td>
<td>{out}</td>
<td>{det}</td>
<td>{sip}</td>
<td>{con}</td>
<td>{dis}</td>
<td>{rege}</td>
</tr>
<tr>
<td>available</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>not available</td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>busy</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>x</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
This Matrix may be used to detect the current handset state and determine the availability. Intended way to determine the detailed state, is to recognize all states with state flow. The status can also be received using OM AXI by request (GetPPState) or notification.

2.1.2.6 XML PP FAC Notification

To notify PP/handset used FACs to an external server an OMP configuration for the “FAC actions” application (see 3.5) has to be done.

2.1.2.7 XML PP Call Completion Notification

To notify PP/handset call completion to an external server an OMP configuration for the “Call Completion” application (see 3.5) has to be done.

2.1.2.8 XML PP Call Park

To notify PP/handset call park hook to an external server an OMP configuration for the “parkCall” application (see 3.5) has to be done.

2.1.2.9 XML PP Call Unpark

To notify PP/handset call unpark hook to an external server an OMP configuration for the “unparkCall” application (see 3.5) has to be done.

2.1.2.10 XML PP Call Pickup

To notify PP/handset call pickup hook to an external server an OMP configuration for the “pickup” application (see 3.5) has to be done.

2.1.2.11 XML PP Call Take

To notify PP/handset call take hook to an external server an OMP configuration for the “take” application (see 3.5) has to be done.

2.1.2.12 XML PP Call Forward

To notify PP/handset call forward hook to an external server an OMP configuration for the “callForward” application (see 3.5) has to be done.

2.1.2.13 XML PP Call Routing

To notify PP/handset call routing hook to an external server an OMP configuration for the “callRouting” application (see 3.5) has to be done.

2.1.2.14 XML PP Call Protection

To notify PP/handset call protection hook to an external server an OMP configuration for the “callProtection” application (see 3.5) has to be done.

2.1.2.15 XML PP Voice Box

To notify PP/handset voice box hook to an external server an OMP configuration for the “voiceBox” application (see 3.5) has to be done.
2.1.2.16 UTF-8 Support

All XML applications have to use the UTF-8 character set. The setting is received in each XML document:

```xml
<?xml version="1.0" encoding="UTF-8"?>
```

For all other settings UTF-8 will be assumed. Note, that this might cause wrong character display at the handset.

2.1.3 XML using HTTP

The OMM-XML interface of release 2.0 is realized as an XML server based on TCP/IP. This means the interface can only react on requests with an XML response. So, events and notifications can only be sent on opened XML-connection established from an external application.

Based on [2] the OMM-XML interface has to be extended to use the HTTP protocol as transport protocol for XML. In case of PBX e. g. for external call lists handling the PBX has to provide for this an HTTP server. The following picture illustrates this for SIP phones:

![XML message flow diagram](image)

Adapting this for the OMM system the XML message flow is shown in the following picture:
2.1.4 XML Client Based Interface

The OMM system has to support an HTML client functionality transporting XML contents. Each GET-request from the OMM contains a URL request and the responses or HTTP-POSTs contain XML-coded display information.

The XML application of the server has to

1. Respond to all “Application” URL-requests with an XML document. The URLs are either configured in the OMM or they are URL links of the current application tree as part of a received XML document.

2. Send an HTTP-POST with an XML document containing an event for a PP to get the current caller list status.

Note: Not forcefully necessary for the OMM, because it can also be done as SIP notification (see chapter 3.4.3.4)

The following MSC illustrates a possible message flow for external call lists.
2.1.5 Protocol Architecture

The following picture illustrates the important protocol stacks used for this use case.
3 Details

3.1 PP And User Interface Requirements

In Mitel SIP-DECT the following requirements are realized for the PP user interface:

- There are some restrictions for list handling at the PP, which is caused in limitations of the DECT stack and the resource of the PP.
  Each list entry is limited to 20 characters and is truncated if longer. Further the PP can only handle lists of maximal 10 entries at a time to browse in. More entries must be handled by the OMM for a line wise scrolling or by the application for an application defined scrolling (e.g. page wise scrolling).

- There are some restrictions for an external XML application to build “TextMenu” XML objects for the DECT handsets:
  1. If the Mitel SIP-DECT system controls the handset scrolling as it is known from the DECT handsets, means, a round robin scrolling is used. For this usage the XML application has to provide “TextMenu” XML objects containing all items without previous or next URI references to further objects. The maximum of 50 items for a PC OMM or 30 for an RFP OMM is supported.
  2. If the XML application controls the scrolling in a page or line wise mode the complete new display content has to be provided and is sent to the DECT handset each time the user goes over the upper and lower end of the display. For this usage the XML application has to provide “TextMenu” XML objects with maximal up to 10 entries containing previous or next URI references to further objects. Due to the capacity of the DECT air interface an optimal use is 6 items (1 page) as display lines for the DECT handset. 6 lines will fit on each supported handset. The server has to provide the scrolling for a page or line wise mode with pages (e.g. 6 items /lines) in that sizing.
  3. Note. The PP software (<= 4.0) does not support lists less than 6 items.

- Only the UTF-8 character sets can be used.

- The usable PPs character set depends on the local language setting of the PP

- The usable PPs language setting is sent in the HTTP Get request and can be used by the application. It is up to the server to use this language or the language setting of SIP registration (if available), English as default or another customized setting.
Maximal 100 XML sessions for a PC OMM or 60 for an RFP OMM is supported at a time.

3.1.1 External Call Lists

In Mitel SIP-DECT the following requirements are realized for the PP user interface:

- If an external call list XML applications is configured in the OMM a local PP configuration is provided to exclusively switch between local and external controlled call lists.

  **PP Menu:** Local menu → Settings → List access →
  
  [ Redial List | Caller list ] → [Automatic | Local | PBX]

  The related local lists are the “redial list” for outgoing calls and the “caller list” for incoming calls. The setting switches between a local (PP internal, an external call list administration in the PBX or an automatic setting depending on the OMM→PP setting during a location registration.

- In the external call lists configuration mode the call lists can be looked up in the local PP menu:

  **PP Menu:** Local menu → Info → Redial list “select an entry”

  **PP Menu:** Local menu → Info → Caller list “select an entry”

- In the external call lists configuration mode the call lists can be used in the dialling mode:

  **PP Menu:** Off hook → OK → Menu → OK → Info → Redial list
  → “select an entry”

  **PP Menu:** Off hook → OK → Menu → OK → Info → Caller list
  → “select an entry”

- In the external call lists configuration mode the redial key (upper left key of the PP) can be used for the external redial list:

  **PP Menu:** “Redial Key” → “select an entry”

- In the external call lists configuration mode the call lists can be used in all send message submenus to select a receiver:

  **PP Menu:** Local menu → Text Message → …

  → Info → Redial list “select an entry”

  **PP Menu:** Local menu → Text Message → …

  → Info → Caller list “select an entry”

- In the external call lists configuration mode all call lists can be programmed as optional short keys. These can be:

  Redial
Caller List
The programmed keys can be used to lookup the related list or as short key in the above scenarios.
Example - A programmable key is programmed with the “Caller List” event:

PP Menu: Off hook
→ “Programmed programmable key for the caller list”
→ “select an entry”

- Delete a call list entry is done with the “C” /”clear” key when the related entry is selected. The complete list is deleted, when the “C” /”clear” key is pressed a long time:
  After any delete action the new list is displayed.

- The local lists of the PP are tagged with icons identifying the current read state. This cannot be used for the external list handling. External list entries are in text form always. The handset can interpret some text as icons to make the content more meaningful. For available icons see the table in 4.1.
  Details of an entry can be retrieved by selecting the entry, when a list was looked up.
  See also the special handling of ‘+’ and ‘-’ character in the first column of a text menu prompt (see 3.4.3.1).

- Asynchronous events for a PP are displayed immediately in a pop up window in idle mode. This can only be done for missed calls.
  The caller list is also affected by asynchronous events to display the current status when entering the “information menu”. The handset needs current caller list state information about “number of new calls” and “number of elements in the caller list” to handle the caller list state in the “info” menu. This must be a notify information from the PBX.
  Additionally the “forwarded terminal” information the PBX has to provide as notification to control the PPs forwarding LED.
  All information are provided by the “PhoneStatus” XML object

3.1.2 XML User Presence
In Mitel SIP-DECT the following requirements are realized for the PP user interface:

- If a user presence XML application is configured in the OMM the local PP menu supports an entry for the related application:
  PP Menu: Local menu → Presence”
Activating this entry the PP requests the OMM to retrieve the main entry of the user presence XML application from the server.

The presence XML application can also be set (programmed) to the programmable keys (“shortcut” keys) at the upper left side of the handset for a faster access.

3.1.3 XML Server Application Menu

In Mitel SIP-DECT the following requirements are realized for the PP user interface:

- If an external XML server application menu is configured in the OMM the PP can access the XML server application menu via the PPs OMM menu.
  
  **PP Menu: Local menu → System menu → Server menu.**

  The OMM menu can also directly be reached by long pressing the local menu key.

- The server application menu can also be used even when the PP is not assigned to a user (a user is not logged in at the PP). In this case the external server can provide an XML menu to support the “login” procedure for a user. The user would not have knowledge about the “login”-feature access code in this case.

3.1.4 Optional Configurable External XML Applications

In Mitel SIP-DECT the following requirements are realized for the PP user interface:

- If an optional configurable XML application is configured in the OMM the local PP menu supports an entry for the access to enter the optional configurable XML applications.
  
  **PP Menu: Local menu → Applications**

  Up to ten external XML applications are available Each XML application as well as the global entry for all ten XML applications can be configured to the programmable keys (“shortcut” keys) at the upper left side of the handset.

3.1.5 XML PP State Notification

The PP has to indicate the silent charging state to the OMM.

3.1.6 XML PP FAC Notification

Nothing to provide by the PP for FAC notification.

3.1.7 XML Call Completion

Nothing to provide by the PP for FAC notification.

3.1.8 XML Call Park Hook

In Mitel SIP-DECT the call park hook is realized for the PP user interface:
- If an external XML call park application is configured in the OMM the PP can access the call park hook via the PPs local menu.
  
  **PP Menu:** Local menu → Park

- Using this feature the user has to be logged in at the device and be in call state.

### 3.1.9 XML Call Unpark Hook

In Mitel SIP-DECT the call unpark hook is realized for the PP user interface:

- If an external XML call unpark application is configured in the OMM the PP can access the call unpark hook via the PPs local menu.
  
  **PP Menu:** Local menu → Unpark call

- Using this feature the user has to be logged in at the device and be in idle state.

### 3.1.10 XML Call Pickup Hook

In Mitel SIP-DECT the call pickup hook is realized for the PP user interface:

- If an external XML call pickup application is configured in the OMM the PP can access the call pickup hook via the PPs local menu.
  
  **PP Menu:** Local menu → Pickup

- Using this feature the user has to be logged in at the device and be in idle state.

### 3.1.11 XML Call Take Hook

In Mitel SIP-DECT the call take hook is realized for the PP user interface:

- If an external XML call take application is configured in the OMM the PP can access the call take hook via the PPs local menu.
  
  **PP Menu:** Local menu → Take

- Using this feature the user has to be logged in at the device and be in idle state.

### 3.1.12 XML Call Forward Hook

In Mitel SIP-DECT the call forward hook is realized for the PP user interface:

- If an external XML call forward application is configured in the OMM the PP can access the call forward hook via the PPs local menu.
  
  **PP Menu:** Local menu → Call diversion

- Using this feature the user has to be logged in at the device and be in idle state.

### 3.1.13 XML Call Routing Hook

In Mitel SIP-DECT the call routing hook is realized for the PP user interface:
• If an external XML call routing application is configured in the OMM the PP can access the call routing hook via the PPs local menu.

   **PP Menu: Local menu → Call routing**

• Using this feature the user has to be logged in at the device and be in idle state.

3.1.14 XML Call Protection Hook

In Mitel SIP-DECT the call protection hook is realized for the PP user interface:

• If an external XML call protection application is configured in the OMM the PP can access the call protection hook via the PPs local menu.

   **PP Menu: Local menu → Call protection**

• Using this feature the user has to be logged in at the device and be in idle state.

3.1.15 XML Voice Box Hook

In Mitel SIP-DECT the voice box hook is realized for the PP user interface:

• If an external XML voice box application is configured in the OMM the PP can access the voice box hook via the PPs Info menu.

   **PP Menu: Local menu → Info → Voice box**

• Using this feature the user has to be logged in at the device and be in idle state.

3.2 OMM Requirements

In the Mitel SIP-DECT the following requirements are realized for supporting external XML applications:

• HTTP/S client support to retrieve XML-documents from the PBX by using a URL request (“http(s)://< web-server>/page”).

• GET HTTP request using the « Accept-Language » tag in the HTTP header as defined for the Mite SIP phones in [2] (chapter 5.15). Other tags of [2] are optional and not relevant.

• HTTP get response header tags for refreshing the currently displayed screen object, in order to request a new URI after a timeout. This refresh tag contains a timeout and a URI to be called (see also [2] chapter 5.14)

• Not used:

   Support of HTTP POST to process asynchronous events notified by the server for the OMM. After receiving an HTTP POST (containing a URL request) the XML caller list status document (“PhoneStatus” XML object) is requested with an HTTP GET request from the server. Optionally the HTTP POST can also contain the XML asynchronous event document.
- Extension of the XML Application supporting the external XML applications. In OMM release 2.1 an XML application (OM AXI) is already available for the OMM configuration, monitoring and event notification. For the external XML applications a new application has to be integrated. A precondition for this is an XML client interface with an XML browser based as close as possible to the XML API ([2]). All XML application to be realized even the XML application for external call lists uses this XML browser. The existing XML interpreter can be reused for the new XML application.

- A complete integration of the XML application handling in the PP software would overload the DECT air interface. Therefore the existing protocol between OMM and PP is used and the OMM has to map the XML protocol to the DECT stack and vice versa. This new phone application controls any key hit action regarding the application, do the scrolling, provides the displaying and reacts on PP actions by requesting the key related URL from the XML server.

- The OMM does not control an XML session. This means, it is not up to the OMM to control the XML dataflow and perform specific context actions. All called URI requests are a result of a user action at the PP, where for the first action of an XML session the DECT air connection (between PP and OMM) is established. Therefore the OMM has no intelligence in stacking display contents / XML objects. When the terminate URI request (“Exit”) is called from an XML document context the DECT air connection to the PP has also be disconnected. This can only be recognised by the OMM when an empty HTTP response is received. This HTTP response (200 OK) is used as trigger that the XML session is ended and the DECT air connection has to be closed.

The following “IPPhone” - objects of [2] are supported by the OMM

- “IPPhoneStatus”
- “IPPhoneTextScreen”
- “IPTextMenu”
- “IPPhoneConfiguration”
- “IPPhoneExecute”
- “IPPhoneInputScreen”
The external call lists application support only the first three objects

For each active XML user the OMM only cashes one object at a time.

3.2.1 External Call Lists

In the Mitel SIP-DECT the following requirements are realized for supporting the XML external call lists handling:

- Support of SIP notify to process asynchronous events notified by the PBX for the OMM. This SIP notification contains the XML caller list status document ("PhoneStatus" XML object). This notify occurs when the state of the caller list in the PBX changes. A SIP notify can also be requested with an HTTP GET request from the PBX. The request is answered with an HTTP response (200 OK) and the SIP notification follows.

- An additional OMM Configuration has to be provided to set up a request for "uri xml call lists" URI, as entry to get the redial lists and the caller list for a PP and as well as to get the current caller list state for a PP).

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- port (HTTP / HTTPS port, e. g. 4443)
- optional user and password for HTTPS access
- server IP address and port supporting the call lists
- call lists page (request page for all call lists e. g. "omm.mghc")
  the specific action like "get", "getinfo",
  "delete", etc. must not be configured the are contained in the parameter list.

The OMP and OM AXI support this configuration.

- (SIP) registered PPs might be temporally unreachable because of an empty battery or bad DECT air conditions. If during this time calls is received, missed calls cannot be popped up at the PP either. The OMM acts when the PP is reachable again to request the "uri xml call lists" (with parameter key=33 – see later in the document) and check for the number of new calls and number of entries in the caller list contained in the "PhoneStatus" XML object. This numbers is used to synchronize the PPs caller list states and to send the number for the new calls to pop up the "<n> missed calls" window at the PP.
3.2.2 XML User Presence

Additionally to 3.1.1 the following requirements are realized for supporting the user presence XML application:

- An additional OMM configuration has to be provided to set up a request for “uri xml user presence” URI, as entry to get access the user presence XML application.

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the user presence
- page (request page of the user presence XML application)

The OMP and OM AXI support this configuration.

3.2.3 XMLServer Application Menu

Additionally to 3.1.1 and 3.1.6 the following requirements are realized for supporting the XML server application menu handling:

- An additional OMM configuration has to be provided to set up a request for “uri xml server menu” URI, as entry to get the PPs XML server application menu.

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the server application menu
- page (request page from the server supporting the server application menu)

The OMP and OM AXI support this configuration.

3.2.4 Optional Configurable External XML Applications

Additionally to 3.1.1 and 3.1.6 the following requirements are realized for supporting the optional configurable external XML applications:

- Up to ten additional OMM configurations has to be provided to set up requests for “uri xml appl 1-10” URIs, as entries to get access to the optional configurable external XML applications.

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the external application
3.2.5 XML PP State Notification

The following requirements are realized for supporting the call state notification:

- An additional OMM configuration has to be provided to set up a request for
  “uri xml pp state notify” URI, as entry to notify the call state to an external
  XML application.
  The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the call state notification
  - page (notify page for the XML application)
  The OMP and OM AXI support this configuration.
- The related PP and the different call states are notified as parameters.

3.2.6 XML PP FAC Notification

The following requirements are realized for supporting the FAC notification:

- An additional OMM configuration has to be provided to set up a request for
  “uri xml pp fac notify” URI, as entry to notify the FAC to an external XML
  application.
  The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the FAC notification
  - page (notify page for the XML application)
  The OMP and OM AXI support this configuration.
- The related PP and the FAC numbers are notified as parameters.

3.2.7 XML PP Call Completion Notification

The following requirements are realized for supporting the call completion notification:

- An additional OMM configuration has to be provided to set up a request for
  “uri xml pp call completion notify” URI, as entry to notify the call completion to an external XML application.
  The data to be configured are:
  - protocol selection (HTTP / HTTPS)
optional user and password for HTTPS access
- server IP address and port supporting the call completion notification
- page (notify page for the XML application)

The OMP and OM AXI support this configuration.

- The related PP, source and destination (SIP) addresses are notified as parameters.

3.2.8 XML PP Call Park

The following requirements are realized for supporting the call park:

- An additional OMM configuration has to be provided to set up a request for
  "uri xml pp call park" URI, as entry to notify the call park to an external
  XML application.

  The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the call park
  - page (notify page for the XML application)

  The OMP and OM AXI support this configuration.

- The related PP data are notified as parameters.

3.2.9 XML PP Call Unpark

The following requirements are realized for supporting the call unpark:

- An additional OMM configuration has to be provided to set up a request for
  "uri xml pp call unpark" URI, as entry to notify the call unpark to an
  external XML application.

  The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the call unpark
  - page (notify page for the XML application)

  The OMP and OM AXI support this configuration.

- The related PP data are notified as parameters.

3.2.10 XML PP Call Take

The following requirements are realized for supporting the call take:

- An additional OMM configuration has to be provided to set up a request for
  "uri xml pp call take" URI, as entry to notify the call take to an external XML
application.
The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the call take
- page (notify page for the XML application)
The OMP and OM AXI support this configuration.

- The related PP data are notified as parameters.

3.2.11 XML PP Call Forward

The following requirements are realized for supporting the call forward:

- An additional OMM configuration has to be provided to set up a request for
  “uri xml pp call forward” URI, as entry to notify the call forward to an
  external XML application.
The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the call forward
  - page (notify page for the XML application)
The OMP and OM AXI support this configuration.

- The related PP data are notified as parameters.

3.2.12 XML PP Call Routing

The following requirements are realized for supporting the call routing:

- An additional OMM configuration has to be provided to set up a request for
  “uri xml pp call routing” URI, as entry to notify the call routing to an
  external XML application.
The data to be configured are:
  - protocol selection (HTTP / HTTPS)
  - optional user and password for HTTPS access
  - server IP address and port supporting the call routing
  - page (notify page for the XML application)
The OMP and OM AXI support this configuration.

- The related PP data are notified as parameters.

3.2.13 XML PP Call Protection

The following requirements are realized for supporting the call protection:
• An additional OMM configuration has to be provided to set up a request for “uri xml pp call protection” URI, as entry to notify the call protection to an external XML application.

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the call protection
- page (notify page for the XML application)

The OMP and OM AXI support this configuration.

• The related PP data are notified as parameters.

3.2.14 XML PP Voice Box

The following requirements are realized for supporting the voice box:

• An additional OMM configuration has to be provided to set up a request for “uri xml pp voice box” URI, as entry to notify the voice box to an external XML application.

The data to be configured are:
- protocol selection (HTTP / HTTPS)
- optional user and password for HTTPS access
- server IP address and port supporting the voice box
- page (notify page for the XML application)

The OMP and OM AXI support this configuration.

• The related PP data are notified as parameters.

3.3 PBX Or XML Server Requirements

3.3.1 External Call Lists

The Mitel SIP-DECT system does not act as the same way like SIP phones do. SIP phones are completely controlled via XML and any key action is controlled via XML (see [2] chapter 4). The external call lists handling to be integrated into the Mitel SIP-DECT system only uses a subset of the Mitel SIP phone XML API without any call handling part. As defined in ([2]) the configuration does not include a URI for the list handling in general. The PBX has to realize the following requirements:

• The PBX has to provide a URI as entry to get the external call lists. This URIs has to be configured in the OMM (see above).

As defined in ([2]) an “IPPhoneTextMenu” XML object of the XML protocol can only contain up to 30 entries! A PC-OMM supports up to 50 entries. Also PP can only
handle maximal 10 entries at a time and use scroll key handling to get more than 10 which are provided by the OMM for a line wise scrolling or by the application for an application defined scrolling (e.g. page wise scrolling).

- Outgoing List / redial list: The PBX maintains and provides a “redial” list containing all successful and failed done calls.

- Incoming list / caller list: The PBX maintains and provides an “Incoming” list containing all successful and failed incoming calls.

See also the special handling of ‘+’ and ‘-‘ character in the first column of a text menu prompt (see 3.4.3.1).

- The PBX has to provide with the URI request (“uri xml call lists”) additional options to get detailed call list entry information, to delete call list entries, to delete a complete call list, to get information of the call list status and to exit the call list (leave browser / no further XML information are required).

- The PBX has to send either a SIP notify event or an HTTP POST event each time when the caller list changes for a PP. This is only be done if the PP is (SIP) registered. To have the call state be updated at the PP, the OMM requests for a caller state information after each SIP registration and when the PP is reachable again after a temporarily unavailability. In this case the configured “uri xml call lists” (with key=33) URI is called by the OMM to request for the SIP notify event containing the “PhoneStatus” XML object.

3.3.2 XML User Presence

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml user presence”) as entry to get an entry menu to the user presence XML application.

3.3.3 XML Server Application Menu

The XML server has to realize the following requirements:

- The server has to provide a URI (“uri xml system menu”) as entry to get the server application menu.

3.3.4 Optional Configurable External XML Applications

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml appl 1-10”) as entry to get an entry menu to the user presence XML application.
3.3.5 XML PP State Notification

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp state notify”) as entry to notify the call state occurred at the PP.

3.3.6 XML FAC Notification

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp fac notify”) as entry to notify the FAC used by a PP.

3.3.7 XML Call Completion Notification

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp call completion notify”) as entry to notify the call completion of a PP.

3.3.8 XML Call Park

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp call park”) as entry to provide the application for the call park hook of a PP.

3.3.9 XML Call Unpark

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp call unpark”) as entry to provide the application for the call unpark hook of a PP.

3.3.10 XML Call Pickup

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp call pickup”) as entry to provide the application for the call pickup hook of a PP.

3.3.11 XML Call Take

The XML Server for this application has to realize the following requirements:

- The XML Server has to provide a URI (“uri xml pp call take”) as entry to provide the application for the call take hook of a PP.

3.3.12 XML Call Forward

The XML Server for this application has to realize the following requirements:
• The XML Server has to provide a URI ("uri xml pp call forward") as entry to provide the application for the call forward hook of a PP.

3.3.13 XML Call Routing

The XML Server for this application has to realize the following requirements:

• The XML Server has to provide a URI ("uri xml pp call routing") as entry to provide the application for the call routing hook of a PP.

3.3.14 XML Call Protection

The XML Server for this application has to realize the following requirements:

• The XML Server has to provide a URI ("uri xml pp call protection") as entry to provide the application for the call park protection of a PP.

3.3.15 XML Voice Box

The XML Server for this application has to realize the following requirements:

• The XML Server has to provide a URI ("uri xml pp voice box") as entry to provide the application for the call voice box hook of a PP.

3.4 HTTP Protocol

3.4.1 HTTP Get Request Header

According to [2] the HTTP header has to contain the Accepted-Language of the handset in each request. The supported languages of the handset are:

• en / English
• de / German
• fr / French
• es / Spanish
• it / Italian
• nl / Dutch
• sv / Swedish
• dk / Danish (da is also accepted)
• no / Norwegian
• pt / Portuguese
• fi / Finnish
• ru / Russian
• pl / Polish
• cs / Czech
• sk / Slovenian
• hu / Hungarian
• tr / Turkish
• et / Estonian

If the server does not support the language, the server can use the language associated to the SIP subscription or English as default.

According to the device identification by the MAC address a subscriber number identification is done (DECT phone devices do not have MAC addresses). The HTTP header has to contain a Subscriber-Number setting in each request.

Other header information like User-Agent and X-Aastra-ExpModi are optional or not used.

3.4.2 URI Requests

URI requests are mostly defined by configuration (see OMP) or sent by the XML server within the XML documents. The placeholders within URLs are used by the OMM to substitute parameters when required. When UTF-8 are substituted they have to be translated for the HTTP request into a correct URL coded form. UTF-8 characters are represented in a hexadecimal coded form in this case.

Example for a substitution after an edit window user input (“IPPhoneInputScreen” XML object):
https://10.103.38.11:443/ppxml/index.php?type=inputscreen1&name=äöüö

3.4.2.1 External Call Lists

The following URI requests are used:

• URI XML Call Lists (“uri xml call lists”) – Configured URI to request the XML document for the PPs call lists. This URI is used to get state information of the call list and is used every time a PP enters the “Redial list” or “Caller list” entry in a selection menu and further actions on it

  URI request specification is:
  {prot}:://{ipaddress}:{port}/{page}
  /?key=<id>&amp;na={subsc}&amp;st={status}

  Where configured:
  {prot} protocol used with the PBX (HTTP “http” | HTTPS “https”)
  {ipaddress} IP address of the PBX – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.
  The placeholder name is “SIPProxy”
  {port} port of the PBX
  {page} page application (e. g. “omm.mghc” for external call lists handling)
PBX – OMM defined:

key=<id> action selector / note: /d is no placeholder!
The key parameter is predefined for the A5000 PBX.
The following values explain the usage. Except of key=33 they are transparent for OMM.

fixed, action key call lists selector settings:
key=33 / get phone status

configured, action key call lists selector settings:
key=20 / get caller list
key=18 / get called list

additional XML contained selector settings:
key=4 / exit
key=5 / back
key=7 / details
key=1 / delete
key=34 / delete all
key=16 / page-wise up scrolling
key=17 / page-wise down scrolling

na={subsc} Subscriber number
st={status} Get phone status request
This parameter has to be used when no key=33 parameter shall be used to request a phone status request from the PBX.
It has to be specified in the caller list (not redial list) URL configuration.
st=1 when the client requests a phone status request
st=0 or missing, when no phone status is requested
If no key and st parameter is defined in the configured caller list URL a get phone status update request will not be requested from the PBX by the OMM.

• URIs as links of the received “TextMenu” XML object:

  handset OK – key → selected entry URI
  handset ESC – key → “cancelAction” URI
  handset UP - key → <SoftKey index="16"> URI / “Up”
  handset DOWN – key → <SoftKey index="17"> URI / “Down”
  handset HOOK OFF – key → Delivers “<Dial>” information to PP and
                             <SoftKey index="8"> URI / “Dial”
  handset HOOK ON – key → <SoftKey index="4"> URI / “Exit”
  handset C – key → <SoftKey index="1"> URI / “Delete”
  handset long pressed C – key → <SoftKey index="34"> URI / “Del.All”
  handset long pressed C – key → Delivers “<Dial>” information to PP and
                             <SoftKey index="2"> URI / “Exit”
IF a URI is not contained in the XML document the corresponding key action is ignored.
Note: see 3.4.3.1 the specific handling for pre dialling and direct dialing for further information.

• URIs as links of the received “TextScreen” XML object:

  handset OK – key IF “Exit” soft key and “<Dial>”
                   information is available:
                   → Delivers “<Dial>” information to PP and
                   <SoftKey index="2"> URI / “Exit”
else:
                   → “doneAction” URI
3.4.2.2 XML User Presence

The following URI requests are used:

- **URI XML user presence** ("uri xml user presence") – Configured URI to request the XML document for the PPs user presence. This URI is used to get the user presence menu and is used every time a PP enters the "XML User Presence" in case of an external XML user presence configuration.
  
  URI request specification is:
  
  `{prot}://{ipaddress}:{port}/{page}`

  Where configured:

  - **{prot}** protocol used with the server (HTTP "http" | HTTPS "https")
  - **{ipaddress}** IP address of the server PBX – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.
    
    The placeholder name is "SIPProxy"

  - **{port}** port of the server
  - **{page}** application start page
  - **{subsc}** Subscriber number

- ... - all other URIs as links of the received XML documents following actions in the XML object.

3.4.2.3 XML Server Application Menu

The following URI requests are used:

- **URI XML Server Application Menu** ("uri xml server menu") – Configured URI to request the XML document for the PPs server application menu. This URI is used to get the XML server application menu
  
  URI request specification is:
  
  `{prot}://{ipaddress}:{port}/{page}`

  Where configured:

  - **{prot}** protocol used with the server (HTTP "http" | HTTPS "https")
  - **{ipaddress}** IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a
3.4.2.4 Optional Configurable External XML Applications

The following URI requests are used:

- **URI XML applications** ("uri xml appl1-10") – Configured URI to request the XML document for up to ten optional configurable external XML applications. This URIs are used to get the XML application menus and is used every time a PP enters access an optional configurable external XML application in case of its configuration.

  URI request specification is:

  \[
  \{\text{prot}\}://\{\text{ipaddress}\}:\{\text{port}\}/\{\text{page}\}/?\text{na}={\text{subsc}}&\text{pp}={\text{ppn}}&\text{usr}={\text{uid}}
  \]

  Where configured:

  \[
  \{\text{prot}\} \text{ protocol used with the server (HTTP "http" | HTTPS "https")} \\
  \{\text{ipaddress}\} \text{ IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.} \\
  \{\text{port}\} \text{ port of the server} \\
  \{\text{page}\} \text{ application start page}
  \]

  Server – OMM defined:

  \[
  \text{na}={\text{subsc}} \text{ Subscriber number} \\
  \text{pp}={\text{ppn}} \text{ PP ppn id; OMM internal reference to a PP device} \\
  \text{usr}={\text{uid}} \text{ user id, OMM internal reference to a user}
  \]

- … - all other URIs as links of the received XML documents following actions in the XML object.

3.4.2.5 XML PP State Notification

The following URI requests are used:

- **URI XML state notification** ("uri xml pp state notify") – Configured URI to request the XML document for the PPs state event. This URI is used to notify the current PP status.

  URI request specification is:

  \[
  \{\text{prot}\}://\{\text{ipaddress}\}:\{\text{port}\}/\{\text{page}\}/?\text{na}={\text{subsc}}&\text{pp}={\text{ppn}}&\text{pa1}={\text{sichaon}}&\text{pa2}={\text{sichaoff}}&\text{pa3}={\text{boot}}&\text{pa4}={\text{reg}}&\text{pa5}={\text{dereg}}&\text{pa6}={\text{onho}}&\text{pa7}={\text{offho}}
  \]

  Where configured:

  \[
  \{\text{prot}\} \text{ protocol used with the server (HTTP "http" | HTTPS "https")} \\
  \{\text{ipaddress}\} \text{ IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.} \\
  \{\text{port}\} \text{ port of the server} \\
  \{\text{page}\} \text{ application start page}
  \]

  Server – OMM defined:

  \[
  \text{na}={\text{subsc}} \text{ Subscriber number} \\
  \text{pp}={\text{ppn}} \text{ PP ppn id; OMM internal reference to a PP device} \\
  \text{usr}={\text{uid}} \text{ user id, OMM internal reference to a user}
  \]
pa8={in}&amp;pa9={out}&amp;pa10={det}&amp;pa11={sip}&amp;pa12={con}&amp;pa13={dis}&amp;pa14={rege}

Where configured:

{prot} protocol used with the server (HTTP "http" | HTTPS "https")
{ipaddress} IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.
The placeholder name is "SIPProxy"

{port} port of the server
{page} application start page

Server – OMM defined:
na={subsc} Subscriber number
pp={ppn} PP ppn id; OMM internal reference to a PP device
pa1={sichaon} Silent charging on/ e.g. pa1=1 when this event occurs, else pa1=0
pa2={sichaoff} Silent charging off/ e.g. pa2=1 when this event occurs, else pa2=0
pa3={boot} PP location registration/ e.g. pa3=1 when this event occurs, else pa3=0
pa4={reg} Successful SIP registration/ e.g. pa4=1 when this event occurs, else pa4=0
pa5={dereg} SIP deregistration/ e.g. pa5=1 when this event occurs, else pa5=0
pa6={onho} On-hook/ e.g. pa6=1 when this event occurs, else pa6=0
pa7={offho} Off-hook/ e.g. pa7=1 when this event occurs, else pa7=0
pa8={in} Incoming call/ e.g. pa8=1 when this event occurs, else pa8=0
pa9={out} Outgoing call/ e.g. pa9=1 when this event occurs, else pa9=0
pa10={det} Detach/ e.g. pa10=1 when this event occurs, else pa10=0
pa11={sip} SIP Notify/ e.g. pa11=1 when this event occurs, else pa11=0
pa12={con} Connect/ e.g. pa12=1 when this event occurs, else pa12=0
pa13={dis} Disconnect/ e.g. pa13=1 when this event occurs, else pa13=0
pa14={rege} Registration event/ e.g. pa14=1 when this event occurs, else pa14=0

An empty HTTP response is required. If non empty HTTP response is received this is handled and affects an existing XML session. For more details see 3.4.2.8 and especially the “allowDestroyAndReplaceSession” document element.

3.4.2.6 XML PP FAC Notification

The following URI requests are used:

- **URI XML FAC Notification (“uri xml pp fac notify”)** – Configured URI to request the XML document for the PPs fac usage. This URI is used to notify the used FAC by the PP.

  URI request specification is:

  `{prot}://(ipaddress):(port):(page)/?na={subsc}&amp;pp={ppn }&amp;fac={fac}`

  Where configured:

  {prot} protocol used with the server (HTTP "http" | HTTPS "https")
  {ipaddress} IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the
current used call server IP address.
The placeholder name is “SIPProxy”

{port} port of the server
{page} application start page
Server – OMM defined:
na={subsc} Subscriber number
pp={ppn} PP ppn id; OMM internal reference to a PP device
fac={fac} used FAC at the PP

An empty HTTP response is required.

3.4.2.7 XML PP Call Completion Notification

The following URI requests are used:

- **URI XML Call Completion Notification** (“uri xml pp call completion notify”) – Configured URI to request the XML document for the PPs call completion usage. This URI is used to notify the completion of a PP.
  
  URI request specification is:
  
  `{prot}://{ipaddress}:{port}/{page}/?na={subsc}&pp={ppn}

Where configured:

{prot} protocol used with the server (HTTP "http" | HTTPS "https")
{ipaddress} IP address of the server – the PBX as SIP call server can have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address.

The placeholder name is “SIPProxy”

{port} port of the server
{page} application start page
Server – OMM defined:
na={subsc} Subscriber number
pp={ppn} PP ppn id; OMM internal reference to a PP device
src={sipsrc} call completion SIP source address
dest={sipdest} call completion SIP destination address

An empty HTTP response is required.

3.4.2.8 XML PP Call Control Hooks

The following URI requests are used:

- **URI XML Call Park** (“uri xml pp call park” / “uri xml pp call unpark” / “uri xml pp call pickup” / “uri xml pp call take” / “uri xml pp call forward” / “uri xml pp call routing” / “uri xml pp call protection” / “uri xml pp voice box”) – Configured URI to request the XML document for the PPs call control (park, unpark, pickup, take, forward, routing, protection, voice box) hook. These URLs are called to use the related XML application.
  
  URI request specification is:
  
  `{prot}://{ipaddress}:{port}/{page}/?na={subsc}&amp;pp={ppn}

Where configured:

{prot} protocol used with the server (HTTP "http" | HTTPS "https")
{ipaddress} IP address of the server – the PBX as SIP call server can
have different IP addresses (primary, secondary or tertiary) depending on current network situations. Therefore a synonym can be used as a placeholder to be replaced by the current used call server IP address. The placeholder name is “SIPProxy”

{port} port of the server
{page} application start page
Server – OMM defined:
na={subsc} Subscriber number
pp={ppn} PP ppn id; OMM internal reference to a PP device

3.4.3 XML-Documents

For the external call lists handling only the “TextMenu”, the “TextScreen” and the “PhoneStatus” XML objects of [2] are used.

For other XML applications The Mitel SIP-DECT system also supports the “InputScreen(single)” XML objects of [2]. All other objects are ignored.

3.4.3.1 “TextMenu” XML Object Usage

The following XML document objects of the “TextMenu” XML object are used (position in parenthesis):

- **IPPhoneTextMenu** (Root tag) – Mandatory root object.
- **defaultIndex** (Root tag) – used.
- **style** (Root tag) – not supported, “numbered” is used by the OMM.
- **destroyOnExit** (Root tag) – not supported, “yes” is used by the OMM.
- **cancelAction** (Root tag) – used to call on exit.
- **allowDestroyAndReplaceSession** (Root tag) – used – if this element is present in this object an existing session will be cleared before this object is pushed to the handset. This is especially used within a SIP notify or in an HTTP responses (e.g., as answer to an XML PP FAC Notification) which normally is empty to push this object content into the handset independently to the current handset state.
- **GoodbyeLockInURI** (MenuItem body) – used – when this element is present, the onHook key is mapped to end the XML session as well as to end an existing call. The PP goes into idle state.
- **wrapList** (Root tag) – not supported, “no” is used by the OMM.
- **Beep** (Root tag) – not supported, “no” is used by the OMM.
- **Timeout** (Root tag) – used to exit the displayed window. This maximal timeout of user inactivity at the handset is used in the OMM by calling the URI specified in the “Softkey exitURI” and close the connection to the handset. Cache of windows is not supported in the OMM.
- **LockIn** (Root tag) – not supported, “no” is used by the OMM.
- `allowAnswer` (Root tag) – not supported, “no” is used by the OMM.
- `allowDrop` (Root tag) – not supported, “no” is used by the OMM.
- `allowXfer` (Root tag) – not supported, “no” is used by the OMM.
- `allowConf` (Root tag) – not supported, “no” is used by the OMM.
- `scrollConstrain` (Root tag) – not supported, “no” is used by the OMM.
- `numberLaunch` (Root tag) – not supported, “no” is used by the OMM.
- `Title` (Body) – used to display as headline by the OMM.
- `wrap` (Title tag) – not supported by the OMM, is ignored.
- `Menuitem` (Body) – used e.g. as call list entry / maximal the number of 50 entries can be used instead of 30
- `Base` (Menuitem tag) – used as Prefix for URI.
- `Icon` (Menuitem tag) – supported by the OMM.
- `Prompt` (Menuitem body) – text to display for this entry.

**Note:** The first character may be handled in a specific way:
For character ‘+’ a checkbox with an icon is generated at the end of line. For character ‘-’ an empty checkbox is generated at the end of line. This can be used e.g. for the caller list to indicate whether the call is answered or not.
To use the checkbox for an entry containing the ‘+’ at first character, two ‘+’ characters have to be used (e.g.: “+4917647114711”).
If the usage of checkboxes should not be used, a blank has to be used before the ‘+’ or ‘-’ character (e.g. “ +4917647114711”), blanks at entry start will be removed by SIP-DECT.

- `URI` (Menuitem body) – used for further operations, e.g. details of a list entry
- `Dial` (Menuitem body) – used as predial information for the next object (e.g. details / TextScreen object) when “OK” key is pressed at the handset
  or used as information for the dial editor at the handset when the URI in this menu item is empty and “OK” is pressed at the handset (XML session is ended)
  or used as information for direct dialling when the “offHook” key is pressed at the handset (XML session is ended)

- `directDial` (Menuitem body) – used when “Dial” is not present.
  Used as direct dial information for the next object (e.g. details / TextScreen object) when “OK” key is pressed at the handset
  or used as information for direct dialling when the URI in this menu item is empty and “OK” is pressed at the handset (XML session is ended)
  or used as information for direct dialling when the “offHook” key is pressed at the handset (XML session is ended).

(additional (add on to [2]) “TextMenu” XML object element used by the OMM)

- `line` (Dial tag) – not supported by the OMM, is ignored.
• **Selection** (MenuItem body) – not supported by the OMM, is ignored.

• **IconList** (Body) – supported by the OMM.

• **Icon** (IconList body) – supported by the OMM (see 4.1).

• **Index** (Icon tag) – supported by the OMM.

• **Softkey** (Body) – used for delete a list entry, delete the complete call list, up/down scrolling exit and dial actions.

• **Index** (Softkey body) – soft key index - Supported indices by the A5000 OMM only:
  - “index=”2” mapped to ESC key of the PP
  - “index=”4” mapped to Up arrow key of the PP
  - “index=”5” mapped to Down arrow key of the PP
  - “index=”8” mapped to C key of the PP
  - “index=”7” mapped to long pressed C-key of the PP

• **Label** (Softkey body) – label expected for the related operation Supported labels by the OMM:
  - “Exit” mapped to “ESC” key of the handset
  - “Dial” mapped to “offHook” key of the handset for direct dialling or mapped to “OK” key of the handset to open the dial editor when no URI in menu item body and no “preDial” softkey is present
  - “preDial” mapped to “OK” key of the handset to open the dial editor when no URI in menu item body is present if present the “Dial” SoftKey is ignored for “OK”. (additional (add on to [2]) “TextMenu” XML object element used by the OMM)
  - “directDial” mapped to “offHook” key of the handset to setup the call directly if present the “Dial” SoftKey is ignored for “offHook”. (additional (add on to [2]) “TextMenu” XML object element used by the OMM)
  - “Up” mapped to “Up arrow” key of the handset
  - “Down” mapped to “Down arrow” key of the handset
  - “Delete” mapped to “C” key of the handset
  - “Delete All” mapped to long pressed “C”-key of the handset

• **URI** (Softkey body) – URI call to process the related action.

Example:
3.4.3.1.1 “GET CALL LIST” XML – RESPONSE

As response to the get call list URI request the server has to send the current call list as result. The number of entries is limited to the number in the request. In any case of an error the server responds with an HTTP error (404) “page not found”.

Response example after get caller list:

```
<IPPhoneTextMenu
  defaultIndex="1"
  destroyOnExit="yes"
  cancelAction=""
  style="numbered"
  Beep="no"
  Timeout="20"
  LockIn="no"
  allowAnswer="no"
  allowDrop="no"
  allowXfer="no"
  allowConf="no"
  wrapList="no">
  <Title wrap="no">Caller list</Title>
  <Prompt>3:17pm K.Winslet</Prompt>
  <URI>https://{ipaddress}:4443/omm.mghc/?key=7&na=4711&selection=441-1</URI>
</IPPhoneTextMenu>
```
Note: The server has to prepare the “prompt” text to display at the PP for each entry! For the OMM this text is transparent and displayed at the PP as received. The content is not wholly analysed or processed by the OMM. Only truncation is done when necessary.

To delete an entry the URI has to be build by the “Softkey-Delete URI” and the selected entry URI.

The red marked parts in the example are optional. If missing the related configuration settings are used.

3.4.3.1.2 “DELETE CALL LIST” (ENTRY) XML – RESPONSE

As response to the delete call list URI request the server has to send the complete new updated call list as result. In any case of an error the server responds with an HTTP error (404) “page not found”.

Response examples after deleting the complete caller list:

Result list is empty:

```xml
<IPhoneTextScreen
destroyOnExit="yes"
cancelAction=""
Beep="no"
Timeout="20"
LockIn="no"
allowAnswer="no"
allowDrop="no"
allowXfer="no"
```
allowConf="no"
allowDTMF="no"
scrollUp=""
scrollDown=""
scrollLeft=""
scrollRight="">
  <Title wrap="no">Caller List</Title>
  <Text>Caller list is empty</Text>
</IPhoneTextScreen>

Result list is not empty:

<IPhoneTextMenu
defaultIndex="0"
destroyOnExit="yes"
style="numbered"
Beep="no"
Timeout="20"
LockIn="no"
allowAnswer="no"
allowDrop="no"
allowXfer="no"
allowConf="no"
cancelAction=""
wrapList="no">
  <Title wrap="no">Caller List</Title>
  <MenuItem base="">
    <Prompt>3:18pm J.Lennon</Prompt>
    <URI>https://{ipaddress}:4443/omm.mghc.getinfo/?key=7
        &amp;na=4711&amp;selection=xxx</URI>
    <Dial>5462</Dial>
    <Selection>xxx</Selection>
  </MenuItem>
  <SoftKey index="1">
    <Label>Delete</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=1</URI>
  </SoftKey>
  <SoftKey index="34">
    <Label>Del.All</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=34</URI>
  </SoftKey>
  <SoftKey index="4">
    <Label>Exit</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
  </SoftKey>
</IPhoneTextMenu>

**Note:** The server has to prepare the “Text”/"prompt” text to display at the PP for as each entry! For the OMM this text is transparent and displayed at the PP as received. The content is not wholly analysed or processed by the OMM. Only truncation is done when necessary
3.4.3.1.3 “Exit Call Lists” XML – Response

In case of SIP Connexion terminals the server answers to the “exit” URI with an “IPPhoneStatus” in the 200 OK http. In case of IP DECT/OMM the “IPPhoneStatus” has only to be sent if the call log has changed otherwise the server has to send an empty 200 OK http response.

3.4.3.2 “TextScreen” XML Object Usage

The “TextScreen” XML object can either be contained in a SIP notification/HTTP Post event or can requested by a URI within an XML object.

The following XML document objects of the “TextScreen” XML object are used (position in parenthesis):

- **IPPhoneTextScreen** (Root tag) – Mandatory root object.
- **destroyOnExit** (Root tag) – not supported, “yes” is used by the OMM.
- **cancelAction** (Root tag) – used, e. g. for questions = no.
- **doneAction** (Root tag) – used, e. g. for questions = yes.
- **allowDestroyAndReplaceSession** (Root tag) – used – if this element is present in this object an existing session will be cleared before this object is pushed to the handset. This is especially used within a SIP notify or in an HTTP responses (e. g. as answer to an XML PP FAC Notification) which normally is empty to push this object content into the handset independently to the current handset state.
- **GoodbyeLockInURI** (MenuItem body) – used – when this element is present, the onHook key is mapped to end the XML session as well as to end an existing call. The PP goes into idle state.
- **wrapList** (Root tag) – not supported, “no” is used by the OMM.
- **Beep** (Root tag) – not supported, “no” is used by the OMM.
- **Timeout** (Root tag) – used to exit the display window. This maximal timeout of user inactivity at the handset is used in the OMM by calling the URI specified in the “Softkey exitURI” and close the connection to the handset. Cache of windows is not supported in the OMM.
- **LockIn** (Root tag) – not supported, “no” is used by the OMM.
- **allowAnswer** (Root tag) – not supported, “no” is used by the OMM.
- **allowDrop** (Root tag) – not supported, “no” is used by the OMM.
- **allowXfer** (Root tag) – not supported, “no” is used by the OMM.
- **allowConf** (Root tag) – not supported, “no” is used by the OMM.
- **allowDTMF** (Root tag) – not supported, “no” is used by the OMM.
- **screenUp** (Root tag) – not supported, “no” is used by the OMM.
- **screenDown** (Root tag) – not supported, “no” is used by the OMM.
- **screenLeft** (Root tag) – not supported, “no” is used by the OMM.
- **screenRight** (Root tag) – not supported, “no” is used by the OMM.
- **Title** (Body) – used to display as headline by the OMM.
- **wrap** (Title tag) – not supported by the OMM, is ignored.
- **Text** (Body) – used as display text
  - **Dial** (Body) – used as information for the dial editor at the handset when the
    “OK” is pressed at the handset (XML session is ended)
    or used as information for direct dialling when the “offHook” key
    is pressed at the handset (XML session is ended)
  (additional (add on to [2]) “TextScreen” XML object element used by the OMM)
- **directDial** (Body) – used when “Dial” is not present.
  Used as information for direct dialling when “OK” is pressed at the
  handset (XML session is ended)
  or used as information for direct dialling when the
  “offHook” key is pressed at the handset (XML session is ended).
  (additional (add on to [2]) “TextMenu” XML object element used by the OMM)
- **Softkey** (Body) – used for exit / delete the currently used entry
  (e. g. when called from textmenu object) and dial action.
- **Index** (Softkey body) – soft key index
- **Label** (Softkey body) – label expected for the related operation
  Supported labels by the OMM:
  “Exit” mapped to “ESC” key of the handset
  “Dial” mapped to “offHook” key of the handset for direct dialling
  or mapped to “OK” key of the handset to open the
dial editor when no “preDial” softkey is present
  “preDial” mapped to “OK” key of the handset to open the dial
editor
  If present the “Dial” SoftKey is ignored for “OK”.
  (additional (add on to [2]) “TextMenu” XML object
element used by the OMM)
  “directDial” mapped to “offHook” key of the handset to setup the
call directly
  If present the “Dial” SoftKey is ignored for “offHook”.
  (additional (add on to [2]) “TextMenu” XML object
element used by the OMM)
  “Delete” URI to call when “C” key is pressed at the handset
- **URI** (Softkey body) – URI call to process the related action.

Example:
3.4.3.2.1 XML "TEXTSCREEN" RESPONSE / GOT AS CALL LIST DETAILS

As response to the URI request initiated by a selection for call list entry the server has to send the detailed information of this entry as result. In any case of an error the server responds with an HTTP error (404) “page not found”.

Response example for a detailed call list entry request:

```xml
<AstraIPPhoneTextScreen
destroyOnExit="yes/no"
cancelAction="some URI"
doneAction="some URI"
Beep="yes/no"
Timeout="some integer"
allowAnswer="yes/no"
allowDrop="yes/no"
allowXfer="yes/no"
allowConf="yes/no"
LockIn="yes/no"
allowDTMF="yes/no"
scrollUp="some URI"
scrollDown="some URI"
scrollLeft="some URI"
scrollRight="some URI">
  <Title wrap="yes/no">Caller Info</Title>
  <Text>Date: 10/07/20 3:17:10pm
  Number: 5461
  Type: office
  Name: Winslet
</AstraIPPhoneTextScreen>
```
First name: Kate
Restriction: no
Status: answered
Unanswered: 5
Missed: 2

<Dial>5462</Dial>
<SoftKey index="1">
  <Label>Delete</Label>
  <URI>https://{ipaddress}:4443/omm.mghc/?key=1</URI>
</SoftKey>
<SoftKey index="4">
  <Label>Exit</Label>
  <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
</SoftKey>

Note: The server has to prepare the detailed information “text” text to display at the PP! For the OMM this text is transparent and displayed at the PP as received. Only truncation is done when necessary. The content is not analysed or processed by the OMM. Newline is also transparent and reflects to a new line in the PPs display.
To delete the entry of the “Softkey-Delete URI” is used.
As result of a PPs “ESC” action the cancel action URI is called (might be used as “no”-result for questions shown at the PPs display).
As result of a PPs “OK” action the done action URI is called (might be used as “yes”-result for questions shown at the PPs display).
The red marked parts in the example are optional. If missing the related configuration settings are used.

3.4.3.3 “PhoneStatus” XML Object Usage

The “PhoneStatus” XML object is contained in a SIP notification/HTTP Post (not supported by the OMM) event.

The following XML document objects of the “PhoneStatus” XML object are used (position in parenthesis):

- **IPPhoneStatus** (Root tag) – Mandatory root object.
- **Beep** (Root tag) – not supported, “no” is used by the OMM.
- **triggerDestroyOnExit** (Root tag) – not supported, “yes” is used by the OMM.
- **Session** (Root tag) – not supported by the OMM.
- **Message** (Body) – used as related information of the used index
- **Index** (Message body) – index="0" ➔ handset/PP forwarded manages the ICON (call deflected/call forwarding) in the very upper status line of the handset
  index="1" ➔ do not disturb
  index="2" ➔ number of new calls (all unanswered calls in the caller list since the last caller list look up / this value is used to pop up at the PP
display in idle state)
index="3" → dual homing
index="4" → locked handset
index="5" → number of caller list entries
index="6" → handset/PP parked

manages the ICON (call parked) in
the idle line of the handset

// Bold marked are supported by the OMM for
certain actions at the PP
All others are just used to display the message in
the PP’s status line. Note that the number of
characters is limited.
Note: This is not visible at the handset!

- **Type** (Message body) – not supported by the OMM.
- **Timeout** (Message body) – not supported by the OMM.

Example:

```xml
<AstraIPPoneStatus
  Beep = “yes/no”
  triggerDestroyOnExit = “yes/no”
>
  <Session>Session ID</Session>
  <Message
    Index = “index”
    Type = “alert”
    Timeout = “timeout”
  >Message</Message>
  <!---Additional Message Items may be added -->
</AstraIPPoneStatus>
```

As response to the call list state info URI request the server has to send the caller list
information as result. In any case of an error the server responds with an HTTP error
(404) “page not found”.
Response example after requesting for “PhoneStatus”:

```xml
<IPPhoneStatus
  triggerDestroyOnExit="yes"
  Beep="no">
  <Session>Session ID</Session>
  <Message index="0"> Forwarded </Message>
  <Message index="1"> 1 </Message>
  <Message index="2"> 25 </Message>
  <Message index="4"> call parked </Message>
</IPPhoneStatus>
```

**Note**: The server has to prepare the “Message” text as a number! For the OMM this number
is used to synchronize the PP caller list state.

For the Boolean attributes (e. g. index="0") the “Message” text is ignored. The OMM
interprets any message text as “true” when the index is present. Otherwise the
OMM interprets the attribute as “false”.
In case of ‘\texttt{index=5} \rightarrow \text{number of caller list entries}’ or ‘\texttt{index=2} \rightarrow \text{number of new calls}’ all elements (\texttt{index=0}, \texttt{index=5} and \texttt{index=2}) must be set in one IPPhoneStatus object to update the state in the handset.

3.4.3.4 “InputScreen (single)” XML Object Usage

The “TextScreen” XML object can either be contained in a SIP notification/HTTP Post (not supported by the OMM) event or can be requested by a URI within an XML object.

The following XML document objects of the single “InputScreen” XML object are used (position in parenthesis):

- \texttt{IPPhoneInputScreen} (Root tag) – Mandatory root object.
- \texttt{type} (Root tag) – used, specifies the kind of input.
- \texttt{password} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{editable} (Root tag) – used to make the field editable.
- \texttt{destroyOnExit} (Root tag) – not supported, “yes” is used by the OMM.
- \texttt{cancelAction} (Root tag) – used, e.g. for questions = no.
- \texttt{allowDestroyAndReplaceSession} (Root tag) – used – if this element is present in this object an existing session will be cleared before this object is pushed to the handset. This is especially used within a SIP notify or in an HTTP responses (e.g. as answer to an XML PP FAC Notification) which normally is empty to push this object content into the handset independently to the current handset state.
- \texttt{GoodbyeLockInURI} (MenuItem body) – used – when this element is present, the onHook key is mapped to end the XML session as well as to end an existing call. The PP goes into idle state.
- \texttt{Beep} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{Timeout} (Root tag) – used to exit the display window. This maximal timeout of user inactivity at the handset is used in the OMM by calling the URI specified in the “Softkey exitURI” and close the connection to the handset. Cache of windows is not supported in the OMM.
- \texttt{LockIn} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{inputLanguage} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{allowAnswer} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{allowDrop} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{allowXfer} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{allowConf} (Root tag) – not supported, “no” is used by the OMM.
- \texttt{Title} (Body tag) – used to display as headline by the OMM.
- **Wrap** (Title tag) – not supported, “no” is used by the OMM.
- **Prompt** (body) – used as guidance.
- **URL** (body) – used after user input completion.
- **Parameter** (body) – used parameter in the URI request.
- **Default** (body) – used to display the default value.
- **Softkey** (Body) – used for exit, dial and submit actions.
- **Index** (Softkey body) – soft key index –
- **Label** (Softkey body) – label expected for the related operation

  Supported labels by the OMM:
  - ”Exit” mapped to ESC key of the PP
  - ”Dial” mapped to Off Hook key of the PP
  - ”Submit” mapped to OK-key of the PP

- **URI** (Softkey body) – URI call to process the related action.

Example:

```
<AstraIPPhoneInputScreen
  type = "IP/string/number/timeUS/timeInt/dateUS/dateInt"
  password = "yes/no"
  editable = "yes/no"
  destroyOnExit = "yes/no"
  cancelAction = "some URI"
  Beep = "yes/no"
  Timeout = "some integer"
  allowAnswer = "yes/no"
  allowDrop = "yes/no"
  allowXfer = "yes/no"
  allowConf = "yes/no"
  LockIn = "yes/no"
  inputLanguage = "English/French/German/Italian/Spanish"
  or "Français/Español/Deutsch/Italiano"
>
  <Title wrap="yes/no">Title string</Title>
  <Prompt>Guidance for the input</Prompt>
  <URL>Target receiving the input</URL>
  <Parameter>name of the parameter added to URL</Parameter>
  <Default>Default Value</Default>
</AstraIPPhoneInputScreen>
```

**Note:** For the input type="IP" an alternative input can be used instead of the dot-notation e.g. “255.255.255.0” for an easier input at the handset. The dot-notation input would need a bulky handling between number and ’.’ input! Therefore a ‘#’ is accepted instead of a ‘.’ input and e.g. an input of “255#255#255#0” is automatically changed to “255.255.255.0”.

3.4.3.5  “InputScreen (multiple)” XML Object Usage

- **allowDestroyAndReplaceSession** (Root tag) – used – if this element is present in this object an existing session will be cleared before this object is pushed to the handset. This is especially used within a SIP notify or in an HTTP responses (e.g. as answer to an XML PP FAC Notification) which normally is empty to push this object content into the handset independently to the current handset state.

- **GoodbyeLockInURI** (MenuItem body) – used – when this element is present, the onHook key is mapped to end the XML session as well as to end an existing call. The PP goes into idle state.

3.4.3.6  “PhoneConfiguration” XML Object Usage

The “PhoneConfiguration” XML object is contained in a SIP notification/HTTP Post event.

The following XML document objects of the “PhoneConfiguration” XML object are used (position in parenthesis):

- **IPPhoneConfiguration** (Root tag) – Mandatory root object.
- **Beep** (Root tag) – not supported, “no” is used by the OMM.
- **triggerDestroyOnExit** (Root tag) – not supported, “yes” is used by the OMM.
- **setType** (Root tag) – not supported by the OMM.
- **ConfigurationItem** (Body) – used to set a configuration item at the PP.
- **setType** (ConfigurationItem tag) – “local” is for some specific parameters supported (see below) – otherwise “remote is as assumed.

**Parameter** (ConfigurationItem) – configuration parameter to be changed. Parameters supported by the OMM:

- **“Set system name”** to set/change/delete the system name at the PPs display
- **“Set address”** to set/change/delete the PP address (e.g. dial number) at the PPs display
- **“Set user name”** to set/change/delete the user name at the PPs display
- **“Set idle line”** to set/change/delete the idle display line (e.g. presence state) at the PPs display
- **“Set forwarding”** to set/delete the call forwarding-ICON in the very upper status line of the handset / any value for this parameter sets the “Set language” to set the PP to a certain language.

The following like the HTTP language settings (in chapter 3.4.1) can be used:

- en / English
- de / German
- fr / French
- es / Spanish
- it / Italian
All other settings set the PP language to default language "en" (English).

**Note:** This feature is only available in OMM release 4.0 or later.

- "**Set call protection**" to set/delete the call protection-ICON in the very upper status line of the handset / any value for this parameter sets the call protection icon

  "**sip line1 display name**" used for XML handset provisioning to manage the SIP user display name. ‘setType="local"’ is required

  "**sip line1 user name**" used for XML handset provisioning to manage the SIP user (name/address/number). ‘setType="local"’ is required

- "**sip line1 auth name**" used for XML handset provisioning to manage the SIP user account name. ‘setType="local"’ is required

- "**sip line1 password**" used for XML handset provisioning to manage the SIP user account password. ‘setType="local"’ is required

- **Value** (ConfigurationItem) -- new value of the configuration parameter / an empty value means delete

In case XML handset provisioning configuration setting for a SIP user all parameters ("sip line1 display name", "sip line1 user name", "sip line1 auth name" and "sip line1 password") have to be set in one "PhoneConfiguration" XML Object. Depending on the corresponding value the user can be added, removed or changed in the OMM database. For further information see 0.

Example:
<AstraIPPhoneConfiguration
  triggerDestroyOnExit="yes/no"
  Beep="yes/no"
  setTypeName="remote/local/oneBoot/override">
  <ConfigurationItem setTypeName="remote/local/oneBoot/override">
    <Parameter>parameter</Parameter>
    <Value>value</Value>
  </ConfigurationItem>
</AstraIPPhoneConfiguration>

<iIPPhoneConfiguration
  triggerDestroyOnExit="yes"
  Beep="no">
  <ConfigurationItem>
    <Parameter>set system name</Parameter>
    <Value>New system name</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set user name</Parameter>
    <Value>New user name</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set address</Parameter>
    <Value>New dial number</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set forwarding</Parameter>
    <Value>Forwarding</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set call protection</Parameter>
    <Value>Call Protection</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set idle line</Parameter>
    <Value>New Idle Line</Value>
  </ConfigurationItem>
  <ConfigurationItem>
    <Parameter>set language</Parameter>
    <Value>sv</Value>
  </ConfigurationItem>
</IIPPhoneConfiguration>

3.4.3.7  "PhoneExecute" XML Object Usage

The "PhoneExecute" XML object is contained in a SIP notification/HTTP Post event.

The following XML document objects of the "PhoneExecute" XML object are used (position in parenthesis):

- **IPPhoneExecute** (Root tag) – Mandatory root object.
- **Beep** (Root tag) – not supported, "no" is used by the OMM.
• triggerDestroyOnExit (Root tag) – not supported, "yes" is used by the OMM.
• allowDestroyAndReplaceSession (Root tag) – used – if this element is present in this object an existing session will be cleared before this object is pushed to the handset. This is especially used within a SIP notify or in an HTTP responses (e.g., as answer to an XML PP FAC Notification) which normally is empty to push this object content into the handset independently to the current handset state.

• not supported, is used by the OMM

Example:

```xml
<AstraIPPoneExecute
  Beep = "yes/no"
  triggerDestroyOnExit="yes/no"
>
  <ExecuteItem URI="http://myserver.com/myscript.php" />
</AstraIPPoneExecute>
```

For further possible URI actions see [2] and note the restrictions in 5.

### 3.4.4 SIP Notify Events

Both events have the same XML document as body. There are two kinds of documents that can be received:

• **Event with an XML content** – The following “IPPone” – object events of [2] are supported by the OMM
  - “IPPoneStatus”
  - “IPPoneConfiguration”
  - “IPPoneTextScreen”
  - “IPPoneInputScreen” *(1)*
  - “IPPoneTextScreen” *(1)*
  - “IPTextMenu” *(1)*

*(1)* The objects are only be processed during an established DECT connection / in DECT call state.

Examples:

**IPPoneStatus:**

```
NOTIFY sip:200@10.30.100.103:5060 SIP/2.0
Via: SIP/2.0/UDP 10.30.100.103:5060;branch=z9hG4bK7bbc1fac;rport
From: <sip:201@10.30.100.103:5060>;tag=81be2861f3
To: Jacky200 <sip:201@10.30.100.103:5060>
```
Contact: <sip:201@10.30.100.103>
call-ID: 59638f5d995c9d301
cSeq: 4 NOTIFY
Max-Forwards: 70
Event: Aastra-xml
Content-Type: application/xml
Content-Length: xxx
<?xml version="1.0" encoding="UTF-8"?>
<IPPhoneStatus>
  triggerDestroyOnExit="yes"
  Beep="no"
  <Session>Session ID</Session>
  <Message index="0"> Forwarded </Message>
  <Message index="1"> </Message>
  <Message index="2"> 1 </Message>
  <Message index="4"> </Message>
  <Message index="5"> 25 </Message>
</IPPhoneStatus>

IPPhoneConfiguration:
NOTIFY sip:200@10.30.100.103:5060 SIP/2.0
Via: SIP/2.0/UDP 10.30.100.103:5060;branch=z9hG4bK7bbc1fac;rport
From: <sip:201@10.30.100.103>;tag=81be2861f3
To: Jacky200 <sip:201@10.30.100.103>
call-ID: 59638f5d995c9d301
cSeq: 4 NOTIFY
Max-Forwards: 70
Event: Aastra-xml
Content-Type: application/xml
Content-Length: xxx
<?xml version="1.0" encoding="UTF-8"?>
<IPPhoneConfiguration>
  <ConfigurationItem
    setType="local">
    <Parameter>Set idle line</Parameter>
    <Value>New Idle Line</Value>
  </ConfigurationItem>
</IPPhoneConfiguration>

- **Event without an XML content** – This event contains a URI link. This has to be requested for retrieving the related XML event document which has to be an
  - “IPPhoneStatus”
  - “IPPhoneConfiguration”
  - “IPPhoneTextScreen”
  - “IPPhoneInputScreen” *1)
  - “IPPhoneTextScreen” *1)
  - “IPTextMenu” *1)

*1) The objects are only be possible during an established DECT connection / in DECT call state.
Examples:

Request for an `IPPhoneStatus` document:

```
NOTIFY sip:200@10.30.100.103:5060 SIP/2.0
Via: SIP/2.0/UDP 10.30.100.103:5060;branch=z9hG4bK7bbc1fac;rport
From: <sip:201@10.30.100.103>;tag=81be2861f3
To: Jacky200 <sip:201@10.30.100.103>
Contact: <sip:201@10.30.100.103>
call-ID: 59838f5d995c9d301
CSeq: 4 NOTIFY
Max-Forwards: 70
Event: Aastra-xml
Content-Type: application/xml
Content-Length: xxx
<?xml version="1.0" encoding="UTF-8"?>
<IPPhoneExecute>
  <ExecuteItem URI="https://{ipaddress}:4443/omm.mghc/?key=333&na=5060"/>
</IPPhoneExecute>
```

Request for an `IPPhoneConfiguration` document:

```
NOTIFY sip:200@10.30.100.103:5060 SIP/2.0
Via: SIP/2.0/UDP 10.30.100.103:5060;branch=z9hG4bK7bbc1fac;rport
From: <sip:201@10.30.100.103>;tag=81be2861f3
To: Jacky200 <sip:201@10.30.100.103>
Contact: <sip:201@10.30.100.103>
call-ID: 59838f5d995c9d301
CSeq: 4 NOTIFY
Max-Forwards: 70
Event: Aastra-xml
Content-Type: application/xml
Content-Length: xxx
<?xml version="1.0" encoding="UTF-8"?>
<IPPhoneExecute>
  <ExecuteItem URI="https://{ipaddress}:4443/phoneConfiguration/?na=5060"/>
</IPPhoneExecute>
```

3.5 XML Application Configuration Via OMP

XML applications can be configured as build in XML applications or as user defined XML applications.

For build in XML applications are mapped to internal implemented functions. They can not be renamed or moved. They have to fit to related the related XML applications on the XML server, and at the PP they are used for certain PP functions (e. g. call lists) or they are mapped to related PPs main menu entries (e. g. “Reception”).

User defined XML applications can be configured individually. If at least one user defined XML applications is been configured an “XML applications” menu is provided in PPs main menu to enter the XML applications menu and further to select one of the configured ones. Alternatively each of the maximal 10 configurable user defined XML
application menus can be set as “Application 1..10” to the softkeys at the PPs left side. The application 1..n are transparent to the OMP configuration, where meaningful names for the each function to an XML application can be set.

The XML URL to be called consists not only of the path on a server but also of additional parameters the XML server has to process for the application. These parameters are not part of the OMM implementation, they are variable and have to be configured also for each XML URL. To build a relation from the URL variables to OMM internal used values predefined variable names can be used within a URL configuration. The following variable names are defined:

- **number** - this is a variable placeholder for a users call number the OMM requests for an XML document from the XML server.

Example (e.g. for caller lists):

```plaintext
omm.mghc.start/?key=18&na={number}
```

The XML server uses the `na`-parameter to identify the specific user. They would replace the known parameters at the specified position in the URL. A specific server request could be:

```plaintext
omm.mghc.start/?key=18&na=4711
```

### 3.5.1 Built-in XML Applications

Depending on the OMM and PP software progress built-in XML applications can be configured. The following built-in XML applications are considered:

- External Call lists ("uri xml call lists")
- XML User Presence ("uri xml user presence")
- XML server application menu ("uri xml server menu")
- Actions URIs ("uri xml pp state notify")
- FAC notification ("uri xml pp fac notify")
- Call completion notification ("uri xml call completion notify")
- Call park hook ("uri xml call park")
- Call unpark hook ("uri xml call unpark")
- Call pickup hook ("uri xml call pickup")
- Call take hook ("uri xml call take")
- Call park forward ("uri xml call forward")
- Call park routing ("uri xml call routing")
- Call park protection ("uri xml call protection")
- Call voice box ("uri xml voice box")
- Administration ("uri xml administration") / only for OMM RFP internal XML administration usage / not configurable

The presence of each build in XML application configuration depends on the OMM version and its capability. The capabilities are indicated via the OM AXI interface in the attributes of the “OpenResp” and “LimitsResp”. See [1] for more details.

The following picture shows the OMP configuration for build in XML applications:

Only “HTTP” or “HTTPS” can be selected for “Protocol”.

3.5.2 Additional User Defined XML Applications

Depending on the OMM and PP software progress up to 10 additional user defined XML applications can be configured (e. g. “Presence” XML application). The administrator has to map each user defined XML application to an additional function
The presence of user defined XML application configurations is indicated via the OM AXI interface in the “OpenResp” confirmation (HAVE_... switches / see [1]).

The following picture shows the OMP configuration for additional user defined XML applications build in XML applications:

Up to 10 additional user-defined XML applications can be added or deleted. Only “HTTP” or “HTTPS” can be selected for “Protocol”.
## 4 Appendix

### 4.1 Useable Icons

The following table lists the available icons that can be used on the handsets display:

<table>
<thead>
<tr>
<th>Icon</th>
<th>UTF8-code</th>
<th>Description</th>
<th>Icon name in IconList of TextMenuObject</th>
</tr>
</thead>
<tbody>
<tr>
<td>📞</td>
<td>ee 80 a1</td>
<td>Icon: Radial Symbol</td>
<td>Icon:RadialSymbol</td>
</tr>
<tr>
<td>📞</td>
<td>ee 81 96</td>
<td>Icon: Telephone book</td>
<td>Icon:Book</td>
</tr>
<tr>
<td>💌</td>
<td>ee 80 ae</td>
<td>Icon: Letter open</td>
<td>Icon:EnvelopeOpened</td>
</tr>
<tr>
<td>💌</td>
<td>ee 80 af</td>
<td>Icon: Letter closed</td>
<td>Icon:Envelope</td>
</tr>
<tr>
<td>💌</td>
<td>ee 80 8e</td>
<td>Icon: PC / email / Tape</td>
<td>Icon:Save</td>
</tr>
<tr>
<td>📦</td>
<td>ee 80 ac</td>
<td>Icon: Save</td>
<td>Icon:Save</td>
</tr>
<tr>
<td>🔔</td>
<td>ee 80 99</td>
<td>Icon: Bell</td>
<td>Icon:PhoneRinging</td>
</tr>
<tr>
<td>✔</td>
<td>ee 80 a4</td>
<td>Icon: Check-Mark positive</td>
<td>Icon:CheckPositive</td>
</tr>
<tr>
<td>✔</td>
<td>ee 80 aa</td>
<td>Icon: Check-Mark negative</td>
<td>Icon:CheckNegative</td>
</tr>
<tr>
<td>🔍</td>
<td>ee 81 81</td>
<td>Icon: Filter</td>
<td></td>
</tr>
<tr>
<td>🔴</td>
<td>ee 80 83</td>
<td>Arrow Up</td>
<td>Icon:ArrowUp</td>
</tr>
<tr>
<td>🔵</td>
<td>ee 80 84</td>
<td>Arrow Down</td>
<td>Icon:ArrowDown</td>
</tr>
<tr>
<td>🔵</td>
<td>ee 80 85</td>
<td>Arrow Left</td>
<td>Icon:ArrowLeft</td>
</tr>
<tr>
<td>🔴</td>
<td>ee 80 86</td>
<td>Arrow Right</td>
<td>Icon:ArrowRight</td>
</tr>
<tr>
<td>🔁</td>
<td>ee 80 88</td>
<td>Fox Key</td>
<td>Icon:FoxKey</td>
</tr>
<tr>
<td>👐</td>
<td>ee 80 b0</td>
<td>Locked</td>
<td>Icon:Locked</td>
</tr>
<tr>
<td>🔍</td>
<td>ee 80 bc</td>
<td>Search</td>
<td>Icon:Search</td>
</tr>
<tr>
<td>🔍</td>
<td>ee 80 be</td>
<td>Info</td>
<td>Icon:Information</td>
</tr>
<tr>
<td>🔴</td>
<td>ee 81 a5</td>
<td>Attention</td>
<td>Icon:Warning</td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Icon</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------</td>
<td>---------------</td>
<td></td>
</tr>
<tr>
<td>ee 80 ba</td>
<td>Tip</td>
<td>Icon:Tip</td>
<td></td>
</tr>
<tr>
<td>ee 80 8a</td>
<td>Telbook Privat number</td>
<td>Icon:Home</td>
<td></td>
</tr>
<tr>
<td>ee 80 8b</td>
<td>Telbook Mobile number</td>
<td>Icon:CellPhone</td>
<td></td>
</tr>
<tr>
<td>ee 80 8c</td>
<td>Telbook Business number</td>
<td>Icon:Office</td>
<td></td>
</tr>
<tr>
<td>ee 81 8c</td>
<td>VIP number</td>
<td>Icon:VIP</td>
<td></td>
</tr>
<tr>
<td>ee 80 8d</td>
<td>Telbook Fax number</td>
<td>Icon:Fax</td>
<td></td>
</tr>
<tr>
<td>ee 80 8e</td>
<td>Telbook Email number</td>
<td>Icon:Email</td>
<td></td>
</tr>
<tr>
<td>ee 80 8f</td>
<td>Telbook Name</td>
<td>Icon:Name</td>
<td></td>
</tr>
<tr>
<td>ee 80 95</td>
<td>Presence</td>
<td>Icon:PresenceAvailable</td>
<td></td>
</tr>
<tr>
<td>ee 80 96</td>
<td>Not reachable</td>
<td>Icon:PresenceAbsent</td>
<td></td>
</tr>
<tr>
<td>ee 80 97</td>
<td>Not available</td>
<td>Icon:PresenceBusy</td>
<td></td>
</tr>
<tr>
<td>ee 80 9b</td>
<td>Hook off / pre dial</td>
<td>Icon:PhoneOffHook</td>
<td></td>
</tr>
<tr>
<td>ee 80 9c</td>
<td>Hook on / Release</td>
<td>Icon:PhoneOnHook</td>
<td></td>
</tr>
<tr>
<td>ee 81 b0</td>
<td>Register recall</td>
<td>Icon:Book</td>
<td></td>
</tr>
<tr>
<td>ee 80 92</td>
<td>DTMF</td>
<td>Icon:BookPrivate</td>
<td></td>
</tr>
<tr>
<td>ee 81 82</td>
<td>3-party</td>
<td>Icon:BookCentral</td>
<td></td>
</tr>
<tr>
<td>ee 80 a0</td>
<td>List Incoming call list</td>
<td>Icon:BookPrivate</td>
<td></td>
</tr>
<tr>
<td>ee 80 a1</td>
<td>List Outgoing call list</td>
<td>Icon:BookCentral</td>
<td></td>
</tr>
<tr>
<td>ee 81 96</td>
<td>List Private directory / Directories</td>
<td>Icon:BookPrivate</td>
<td></td>
</tr>
<tr>
<td>ee 81 99</td>
<td>List Central directory</td>
<td>Icon:BookCentral</td>
<td></td>
</tr>
<tr>
<td>ee 81 8c</td>
<td>List VIP</td>
<td>Icon:BookPrivate</td>
<td></td>
</tr>
<tr>
<td>Code</td>
<td>Description</td>
<td>Icon</td>
<td></td>
</tr>
<tr>
<td>----------</td>
<td>----------------------------------</td>
<td>------------------</td>
<td></td>
</tr>
<tr>
<td>ee 81 81</td>
<td>List Filter</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ee 81 81</td>
<td>Call Filtered</td>
<td>Icon:CallFiltered</td>
<td></td>
</tr>
<tr>
<td>ee 80 a1</td>
<td>Call outgoing</td>
<td>Icon:CallOutgoing</td>
<td></td>
</tr>
<tr>
<td>ee 80 99</td>
<td>Call Waiting</td>
<td>Icon:CallWaiting</td>
<td></td>
</tr>
<tr>
<td>ee 80 a7</td>
<td>Call Rejected</td>
<td>Icon:CallRejected</td>
<td></td>
</tr>
<tr>
<td>ee 81 ad</td>
<td>Call SOS</td>
<td>Icon:CallSOS</td>
<td></td>
</tr>
<tr>
<td>ee 80 9d</td>
<td>Call Headset auto answer</td>
<td>Icon:Headset</td>
<td></td>
</tr>
<tr>
<td>ee 80 98</td>
<td>Call Loudspeaker auto answer</td>
<td>Icon:Speaker</td>
<td></td>
</tr>
<tr>
<td>ee 80 9b</td>
<td>Call Hook auto answer</td>
<td>Icon:CallHookAutoAnswer</td>
<td></td>
</tr>
<tr>
<td>ee 80 b8</td>
<td>Call deflected</td>
<td>Icon:Deflected</td>
<td></td>
</tr>
<tr>
<td>ee 80 a3</td>
<td>Call missed</td>
<td>Icon:CallMissed</td>
<td></td>
</tr>
<tr>
<td>ee 80 a4</td>
<td>Call answered</td>
<td>Icon:CallAnswered</td>
<td></td>
</tr>
<tr>
<td>ee 81 95</td>
<td>Call on voice box</td>
<td>Icon:CallOnVoiceBox</td>
<td></td>
</tr>
<tr>
<td>ee 81 ae</td>
<td>Call VIP</td>
<td>Icon:CallVIP</td>
<td></td>
</tr>
<tr>
<td>ee 82 96</td>
<td>Call parked</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4.2 Detailed Examples

4.2.1 Dial From Caller List

XML-Document *1:
<IPhoneTextMenu>
  <Title wrap="no">Caller Info</Title>
  <Menuitem base="">
    <Prompt> 3:17pm K.Winslet</Prompt>
    <URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=xxx</URI>
    <Dial>5461</Dial>
    <Selection>xxx</Selection>
  </Menuitem>
  <Menuitem base="">
    <Prompt> 3:18pm J.Lennon</Prompt>
    <URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=yyy</URI>
    <Dial>5462</Dial>
    <Selection>yyy</Selection>
  </Menuitem>
  ...
  <Menuitem base="">
    <Prompt> 5:21pm S.Fox</Prompt>
    <URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=zzz</URI>
    <Dial>5490</Dial>
  </Menuitem>
</IPhoneTextMenu>
4.2.2 Delete Complete Redial List

PP display:

“IDLE Menu”

PP “redial” key pressed:

URI request:
HTTP Get response:

```xml
<IPPhoneTextMenu>
  <Title wrap="no">Called Info</Title>
  <MenuItem base=""/>
    <Prompt> 3:17pm K.Winslet</Prompt>
    <URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=xxx</URI>
    <Dial>5461</Dial>
    <Selection>xxx</Selection>
  </MenuItem>
  <MenuItem base=""/>
    <Prompt> 3:18pm J.Lennon</Prompt>
    <URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=yyy</URI>
    <Dial>5462</Dial>
    <Selection>yyy</Selection>
  </MenuItem>
  <SoftKey index="1">
    <Label>Delete</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=1</URI>
  </SoftKey>
  <SoftKey index="34">
    <Label>Del.All</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=34</URI>
  </SoftKey>
  <SoftKey index="4">
    <Label>Exit</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
  </SoftKey>
</IPPhoneTextMenu>

PP display:
3:17pm K.Winslet
3:18pm J.Lennon

PP "clear/C" key pressed long:
URI request:

https://{ipaddress}:4443/omm.mghc/?key=34&na=4711

HTTP Get response:

```xml
<IPPhoneTextScreen>
  <doneAction>https://{ipaddress}:4443/omm.mghc/?key=340&amp;na=4711</doneAction>
  <cancelAction>https://{ipaddress}:4443/omm.mghc/?key=341&amp;na=4711</cancelAction>
  <Title wrap="no">Redial list</Title>
  <Text>Delete complete redial list?</Text>
  <SoftKey index="2">
    <Label>Exit</Label>
    <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
  </SoftKey>
</IPPhoneTextScreen>

PP display:
Delete complete redial list?

PP “OK” key pressed:

URI request:

https://{ipaddress}:4443/omm.mghc/?key=341&na=4711

HTTP Get response:

<IPhoneTextScreen
    Timeout="1"
    cancelAction="https://{ipaddress}:4443/omm.mghc/?key=4"
    doneAction="https://{ipaddress}:4443/omm.mghc/?key=4">
    <Title> wrap="no">Redial list</Title>
    <Text>Request accepted!</Text>
    <SoftKey index="2">
        <Label>Exit</Label>
        <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
    </SoftKey>
</IPhoneTextScreen>

PP display:

Request accepted

After 1 second:

URI request:

https://{ipaddress}:4443/omm.mghc/?key=18&na=4711

HTTP Get response:

<IPhoneTextScreen
    Timeout="1"
    cancelAction="https://{ipaddress}:4443/omm.mghc/?key=4"
    doneAction="https://{ipaddress}:4443/omm.mghc/?key=4">
    <Title> wrap="no">Redial list</Title>
    <Text>Redial list empty</Text>
    <SoftKey index="2">
        <Label>Exit</Label>
        <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
    </SoftKey>
</IPhoneTextScreen>

URI request:

https://{ipaddress}:4443/omm.mghc/?key=4&na=4711

HTTP Get response:

200 OK

PP display:

“IDLE Menu”
4.2.3 Delete A Caller List Entry

PP display:

“IDLE Menu”

PP “caller list” key pressed:

URI request:

https://{ipaddress}:4443/omm.mghc/?key=20&na=4711

HTTP Get response:

"IPPhoneTextMenu" defaultIndex="1"<Title wrap="no">Called Info</Title>
   <MenuItem base=""><Prompt> 3:17pm K.Winslet</Prompt><Dial>5401</Dial>
   <URI>omm.mghc/?key=7&amp;na=4711&amp;selection=1</URI>
   <Selection>1</Selection></MenuItem>
   <MenuItem base=""><Prompt> 3:18pm M.Smith</Prompt><Dial>5402</Dial>
   <URI>omm.mghc/?key=7&amp;na=4711&amp;selection=2</URI>
   <Selection>2</Selection></MenuItem>
   <MenuItem base=""><Prompt> 3:55pm J.Lennon</Prompt><Dial>5403</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=3</URL>
<Selection>3</Selection></MenuItem>
<MenuItem base="">
<Prompt> 4:04pm A.Lennox</Prompt><Dial>5404</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=4</URL>
<Selection>4</Selection></MenuItem>
<MenuItem base="">
<Prompt> 4:24pm S.Claus</Prompt><Dial>5405</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=5</URL>
<Selection>5</Selection></MenuItem>
<MenuItem base="">
<Prompt> 6:10pm K.Richard</Prompt><Dial>5406</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=6</URL>
<Selection>6</Selection></MenuItem>
<MenuItem base="">
<Prompt> 6:56pm M.Faithful</Prompt><Dial>5407</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=7</URL>
<Selection>7</Selection></MenuItem>
<MenuItem base="">
<Prompt> 7:23pm J.Cash</Prompt><Dial>5408</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=8</URL>
<Selection>8</Selection></MenuItem>
<MenuItem base="">
<Prompt> 8:34pm J.Rush</Prompt><Dial>5409</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=9</URL>
<Selection>9</Selection></MenuItem>
<MenuItem base="">
<Prompt> 8:54pm G.Estefan</Prompt><Dial>5410</Dial>
<URL>/omm.mghc/?key=7&amp;na=4711&amp;selection=10</URL>
<Selection>10</Selection></MenuItem>
<SoftKey index="1">
<Label>Delete</Label>
<URI>https://{ipaddress}:4443/omm.mghc/?key=1</URI></SoftKey>
<SoftKey index="34">
<Label>Del.All</Label>
<URI>https://{ipaddress}:4443/omm.mghc/?key=34</URI></SoftKey>
<SoftKey index="4">
<Label>Exit</Label>
<URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI></SoftKey>
<SoftKey index="16">
<Label>Up</Label>
<URI>https://{ipaddress}:4443/omm.mghc/?key=16</URI></SoftKey>
<SoftKey index="17">
<Label>Down</Label>
<URI>https://{ipaddress}:4443/omm.mghc/?key=17</URI></SoftKey>
</IPPhoneTextMenu>

PP display :
3:17pm K.Winslet ➲ Cursor
3:18pm M.Smith
3:55pm J.Lennon
4:04pm A.Lennox
4:24pm S.Claus
6:10pm K.Richard
6:56pm M.Faithful
7:23pm J.Cash
8:34pm J.Rush
8:54pm G.Estefan

PP "DOWN" key pressed 10 times:
3:17pm K.Winslet
3:18pm M.Smith
3:55pm J.Lennon
4:04pm A.Lennox
4:24pm S.Claus
6:10pm K.Richard
URI request:

https://{ipaddress}:4443/omm.mghc/?key=17&amp;na=4711

HTTP Get response:

IPPonPhoneTextMenu> defaultIndex="1"><Title wrap="no" >Called Info</Title>
  <MenuItem base=""><Prompt> 9:11pm M.Jackson</Prompt><Dial>5411</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=11</URI><Selection>11</Selection></MenuItem>
  <MenuItem base=""><Prompt> 9:18pm S.Fox</Prompt><Dial>5412</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=12</URI><Selection>12</Selection></MenuItem>
  <MenuItem base=""><Prompt> 9:55pm J.Denver</Prompt><Dial>5413</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=13</URI><Selection>13</Selection></MenuItem>
  <MenuItem base=""><Prompt> 10:04pm H.Bellafonte</Prompt><Dial>5414</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=14</URI><Selection>14</Selection></MenuItem>
  <MenuItem base=""><Prompt> 10:24pm B.Streisand</Prompt><Dial>5415</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=15</URI><Selection>15</Selection></MenuItem>
  <MenuItem base=""><Prompt> 11:10pm T.Hill</Prompt><Dial>5416</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=16</URI><Selection>16</Selection></MenuItem>
  <MenuItem base=""><Prompt> 11:56pm M.Douglas</Prompt><Dial>5417</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=17</URI><Selection>17</Selection></MenuItem>
  <MenuItem base=""><Prompt> 00:23am M.Streep</Prompt><Dial>5418</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=18</URI><Selection>18</Selection></MenuItem>
  <MenuItem base=""><Prompt> 8:34am T.Turner</Prompt><Dial>5419</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=19</URI><Selection>19</Selection></MenuItem>
  <MenuItem base=""><Prompt> 8:54am L.Taylor</Prompt><Dial>5420</Dial><URI>/omm.mghc/?key=7&amp;na=4711&amp;selection=20</URI><Selection>20</Selection></MenuItem>
</IPPonPhoneTextMenu>

PP display:

9:11pm M.Jackson  ← Cursor
9:18pm S.Fox
9:55pm J.Denver
10:04pm H.Bellafonte

63 | © 2015 MITEL Specification
10:24pm B.Streisand
11:10pm T.Hill
11:56pm M.Douglas
00:23am M.Streep
8:34am T.Turner
8:54am L.Taylor

PP “DOWN” key pressed:
9:11pm M.Jackson
9:18pm S.Fox
9:55pm J.Denver
10:04pm H.Bellafonte
10:24pm B.Streisand
11:10pm T.Hill
11:56pm M.Douglas
00:23am M.Streep
8:34am T.Turner
8:54am L.Taylor

PP “C”/“Clear” key pressed:

URI request:
https://{ipaddress}:4443/omm.mghc/?key=17&na=4711&selection=12
HTTP Get response:

<IPPhoneTextScreen
   Timeout="1"
   cancelAction="https://{ipaddress}:4443/omm.mghc/?key=17"
   doneAction="https://{ipaddress}:4443/omm.mghc/?key=17">
   <Title wrap="no">Redial list</Title>
   <Text>Request accepted!</Text>
   <SoftKey index="2">
      <Label>Exit</Label>
      <URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI>
   </SoftKey>
</IPPhoneTextScreen>

PP display:
Request accepted

After 1 second:

URI request:
https://{ipaddress}:4443/omm.mghc/?key=17&na=4711
HTTP Get response:

<IPPhoneTextMenu defaultIndex="2"> <Prompt> 9:11pm M.Jackson </Prompt><Dial>5412</Dial>
   <Menuitem base=""> <Prompt> 9:55pm J.Denver</Prompt> <Dial>5413</Dial>
   <Menuitem base=""> <Prompt> 10:04pm H.Bellafonte</Prompt> <Dial>5414</Dial>
</Menuitem>
</Menuitem>
<Selection>13</Selection></MenuItem>

<MenuItem base=””><Prompt> 10:24pm B.Streisand</Prompt><Dial>5415</Dial><Selection>14</Selection></MenuItem>

<MenuItem base=””><Prompt> 11:10pm T.Hill</Prompt><Dial>5416</Dial><Selection>15</Selection></MenuItem>

<MenuItem base=””><Prompt> 11:56pm M.Douglas</Prompt><Dial>5417</Dial><Selection>16</Selection></MenuItem>

<MenuItem base=””><Prompt> 00:23am M.Streep</Prompt><Dial>5418</Dial><Selection>17</Selection></MenuItem>

<MenuItem base=””><Prompt> 8:34am T.Turner</Prompt><Dial>5419</Dial><Selection>18</Selection></MenuItem>

<MenuItem base=””><Prompt> 8:54am L.Taylor</Prompt><Dial>5420</Dial><Selection>19</Selection></MenuItem>

<MenuItem base=””><Prompt> 9:00pm P.Collins</Prompt><Dial>5421</Dial><Selection>20</Selection></MenuItem>

<SoftKey index=”1”><Label>Delete</Label><URI>https://{ipaddress}:4443/omm.mghc/?key=1</URI></SoftKey>

<SoftKey index=”34”><Label>Del.All</Label><URI>https://{ipaddress}:4443/omm.mghc/?key=34</URI></SoftKey>

<SoftKey index=”4”><Label>Exit</Label><URI>https://{ipaddress}:4443/omm.mghc/?key=4</URI></SoftKey>

<SoftKey index=”16”><Label>Up</Label><URI>https://{ipaddress}:4443/omm.mghc/?key=16</URI></SoftKey>

<SoftKey index=”17”><Label>Down</Label><URI>https://{ipaddress}:4443/omm.mghc/?key=17</URI></SoftKey>

</IPPhoneTextMenu>

PP display :
9:11pm M.Jackson
9:55pm J.Denver
10:04pm H.Bellafonte
10:24pm B.Streisand
10:40pm H.Bellafonte
11:10pm T.Hill
11:56pm M.Douglas
00:23am M.Streep
8:34am T.Turner
8:54am L.Taylor
9:00pm P.Collins

PP “HOOK ON” key pressed:
URI request:
https://{ipaddress}:4443/omm.mghc/?key=4&na=4711
HTTP Get response:
200 OK

PP display :
"IDLE Menu"
5 Restrictions

5.1 XML Concerned Restrictions

Concerning XML applications the Mitel SIP-DECT system has the following restrictions:

- Currently the Mitel SIP-DECT system supports only a reduced set of the PhoneExecute XML object:

  URI="http:..." supported
  URI="Dial:..." supported, uses dial information as input for the called dial editor at the handset
  URI="directDial:..." supported, uses dial information for direct call (without calling the dial editor)
  URI="DialLine:1:..." supported, same as "Dial"
  URI="Softkey:Exit" supported, ends the XML session of the handset
  URI="Key:Goodbye" supported, end the current call at the handset
  URI="Key:Headset" supported, accept an incoming call at the handset
  URI="Key:OnHook" supported, end the current call at the handset
  URI="Key:OffHook" supported, accept an incoming call at the handset
  URI="Led:..." not supported
  URI="Command:SipRegister" supported, forces a reregistration of the SIP device.
  Other commands are not supported!

EDI="UserLogin <usrNum> <pin>" supported, performs a user login at the handset
EDI="UserLogout <usrNum> <pin>" supported, performs a user logout at the handset
EDI="ExecuteNotify <URI>" supported, see below.

The "Login/Logout"-URI is used to offer an external XML application the possibility to provide a user login or logout for a PP device. In case of a user login a user data set (this can also be a user data set on an external user data server) is tried to login to the device the XML session is working on. The result can be notified to the external XML application when the "ExecuteNotify" is present. The notified URI to be called for a user login with a notify URI "EDI="ExecuteNotify http://ber-rd5022/Execute-Notify" looks like:

  http://ber-rd5022/Execute-Notify?result=NOK&info=wrong-pin

or


If the (login/logout) user is provided on an external user data server (an OMM
external user) the server might have an HTTP-DIGEST authentication access for
the users to login or logout. In this case the pin represents the HTTP-DIGEST
password for the user.

5.2 Direct Dial Restrictions

Direct dialling while browsing in XML menus without using the handset’s dial editor is not
supported to video devices. The user always has to use the pre dial editor in this case.

5.3 Upgrading A5000 Call Lists To OMM SIP-DECT 4.0

Differences:

- Only UTF-8 coding
- No further support of "(count)"-parameter due to changed display scrolling
- Changed display scrolling
  / = 7 entries ==> PBX controls complete scrolling with up and down
  / > 7 entries ==> OMM/handset controls complete scrolling without up and down
  Either PP build in scrolling or XML server controlled scrolling can be used!
  Objects with prev or next URIs refers to XML server controlled scrolling
  Objects with more than 7 items refers to PP build in scrolling.
  Both together (> 7 items and prev or next URIs within an IPPhoneTextMenu object)
cannot be supported!
  Object will be cut to 7 items in case of prev or next URIs in IPPhoneTextMenu!
- Configuration of status request parameter ("key=33")? – Open issue -
- No timeout reduction to 115 seconds for IPPhoneTextMenu object (due to handset
ergonomics and IS_21051 timeout reduced to 115 sec in OMM release 2.1).
- No timeout reduction to 120 seconds for IPPhoneTextScreen object (due to handset
ergonomics and IS_21051 timeout reduced to 120 sec in OMM release 2.1).