OBIHAI
technology, inc.

Obihai Technology, Inc.

OBi 500 Series Device Administration Guide

Models:

OBi508 – 8-Port Voice Service Bridge and Terminal Adapter

OBi504 – 4-Port Voice Service Bridge and Terminal Adapter

January 2017

Introduction

Audience

Cloud Service Providers and Managed Service VARS

Note to End Users

End users are highly encouraged to use the OBiTALK web portal to configure and manage their OBi devices. The reason for this is two-fold. One, the major benefits afforded by the OBi devices are available to be turned on, set-up and modified from within the portal. What's more is the application of the functional configuration is controlled by device configuration files tightly integrated with the settings configured by the OBiTALK portal and cloud-based applications managed by Obihai which make possible the device's "plug-n-play" operation.

Where to Go for Help

Obihai has a number of options available to customers who are seeking help regarding their Obihai products.

Obihai Web Site:

1. Obihai Support Web Site: http://www.obihai.com/support.html
   On this web site visitors will find links to the OBiTALK forum, Documents and Downloads, Tools Tips and Tricks as well as an FAQ / Knowledge Base.
2. Enter a Support Request at: http://www.obihai.com/supportTicketForm.php
3. Go to the OBiTALK forum at: www.obitalk.com/forum
4. E-mail the Obihai Support Team at: support@obihai.com

Notational Conventions

An OBi device configuration parameter and value is represented in the style

Group Name::**ParameterName** = `Parameter Value`

Group Name is the heading of the parameter group on the left side panel of the device configuration web page and may contain spaces. When a group heading has more than one level, each level is separated with a -, such as

Services Providers - ITSP Profile A – SIP::

ParameterName is the name of the parameter as shown on the web page and MUST NOT CONTAIN ANY SPACES. `Parameter Value` is the literal value to assign to the named parameter and may contain spaces. **Group Name** or its top level headings may be omitted when the context is clear. Examples:

> **SP1 Service**::**AuthUserName** = `4082224312`

> **ITSP Profile A - SIP**::**ProxyServer** = `sip.myserviceprovider.com`

A placeholder value to be substituted by your the parameter value you will actually use is shown in italics, as in the following example:

> **SP1 Service**::**AuthPassword** = *your-password*

OBIHAI
technology, inc.

Some configuration parameters can assume one of several choices only. The configured value must match exactly the admissible values, case sensitive and especially no extra white spaces before or after the value. Each admissible value is highlighted in this document to emphasize this requirement. For example `GMT-12:00(Int'l Dateline West)` is one of the admissable choices for the **LocalTimeZone** parameter.

Here are some examples how we highlight some SIP syntaxes in this document:

- A SIP header: **Privacy: id**

- A URL parameter: **;user=phone**

Acronyms

RW	Read write-able (parameter values)
R	Read only (parameter values)
UINT	Unsigned interger (0, 1, 2, …)
INT	Integer

Introduction to OBi500 Devices

Built with a high-performance system-on-a-chip platform to ensure high quality voice conversations, OBi500 devices are dedicated systems targeted at the application of voice over IP services. OBi500 devices have high availability and reliability because they are always-on to make or receive calls. With an OBi device, a computer is not required and a computer does not need to be on to talk to people. To get started, all you need is a phone, power and a connection to the Internet.

OBi500 Devices:

Model	VoIP Accounts	OBiTALK Support	Phone Ports	PSTN Line Ports	GigE Ports	Fast Ethernet Ports	USB Ports	Audio Line In	Audio Line Out	PSTN Failover Ports
OBi508	9	Yes	8	0	1**	3**	3	1	1	1*
OBi504	9	Yes	4	0	1**	3**	3	1	1	1*

* For failover during power outage only
** OBi500 can be configured as a router or as a switch. In router mode, the GigE port functions as the WAN port, and the 3 Fast Ethernet ports as 3 LAN switch ports. In switch mode, all 4 ports function as switch ports

OBi500 Devices Are Complemented by Other OBi Products & Services

OBiTALK: A customer portal for device management allowing members to add people and associated OBi endpoints to "circles of trust" such that additional functionality can be shared amongst authorized users. The OBiTALK portal is also where members can download the OBiAPP and OBiON applications for PCs and the iPhone, iPad, iPod touch & Android devices, respectively.

OBiON for iPhone, iPad, iPod touch & Android Devices: An application for iPhone, iPad, iPod touch and Android devices which makes possible placing and receiving calls to/from other OBi endpoints.

OBiAPP for PC: An application for a PC that facilitates placing and receiving calls to/from other OBi endpoints.
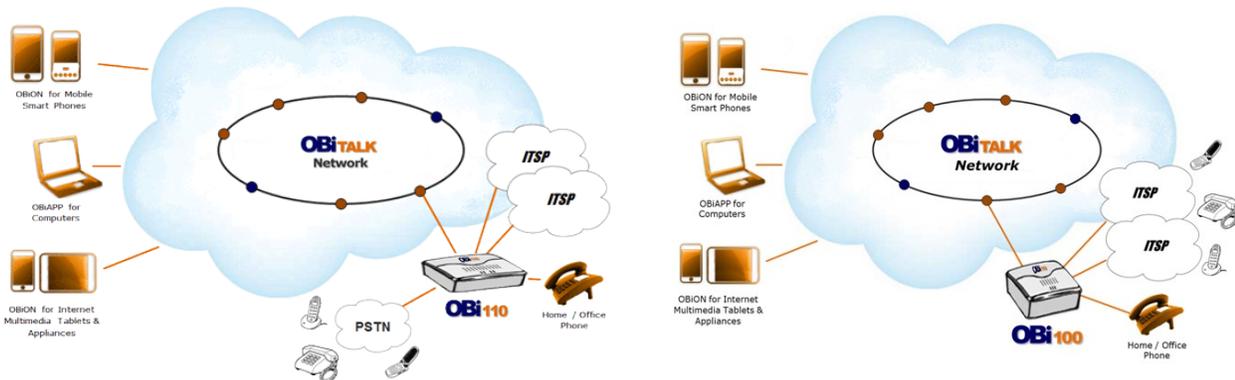
Key Features of the OBi500 Voice Service Bridge / Telephone Adapter:

- Aggregation and Bridging of Eight (8) SIP and/or Google Voice
- Fully configurable multi-level Automatic Attendants
- Flexible and powerful Digit Maps and Call Routing Rules configuration
- PBX-like internal calling features among the built-in FXS ports
- Audio Line In and Line Out ports for music streaming and paging
- Flexible hunt groups configuration
- Virtual stacking

OBiTALK Web Portal Integration

- Configuration and Management of OBi Endpoints
- Download OBi Client Applications for PCs, Mobile Phones & Internet Devices
- Creating & Joining Circles of Trust So You Can Share Your OBi
- Setting Up Your OBi Endpoint Speed Dial Directory

Configurable to Work with Any SIP Compliant Internet Telephone Service or Google Voice Communications Service
Configurable to Work with Most Loop Start Analog Telephone Lines
Analog Phone & Telephone Line Impedance Agnostic

Robust Telephony Features:

- Message Waiting Indication - Visual and Tone Based
- Speed Dialing of 99 OBi Endpoints or Numbers
- Three Way Conference Calling with Local Mixing
- Hook Flash Event Signaling
- Caller ID – Name & Number
- Caller ID Spoofing
- Call Waiting
- Call Forward - Unconditional
- Call Forward on Busy
- Call Forward on No Answer
- Call Transfer
- Call Park
- Anonymous Call
- Block Anonymous Call
- Do Not Disturb
- Call Return
- Repeat Dialing
- Collaboration – PBX-like call features among all phone ports
- Paging through the Line Out port
- Music On Hold
- Multi-level Auto Attendants with fully customizable prompts

OBi508

OBi504

Powerful Call Routing & Voice Service Features:

- SIP Support for Voice and Fax Over IP (T.38 Relay and G711 Passthrough)
- OBiTALK Managed VoIP Network for OBi Endpoint Devices & Applications
- High Quality Voice Encoding Using G.711, G.726, G.729, iLBC Algorithms
- Recursive Digit Maps & Associated Call Routing (Outbound, Inbound)
- Service Continuity in Case of Power or Network Failure (Configurable)

Physical Interfaces of the OBi508

Overview of the OBi508 Physical Interfaces

OBi508 Front and Rear   OBi 504 Front and Rear

Connecting Power to the OBi Device

Connect the supplied 12 volt power adapter to the OBi device and the wall outlet or working power strip. Only use the power adapter supplied with the original packaging to power the OBi device. Use of any power adaptor other than what was provided with the OBi will void the warranty and may cause the unit to not function at all or cause undesired operation.

LED Description and LED Behaviour

There are fourteen (14) LED lights on the front of the OBi508 and ten (10) on the OBi504. They are used to provide the user with a visual indication of the working order and general status of key functional aspects of the OBi device. Under normal operating conditions the LEDs should show green (solid or blinking) signals.

Here below, are specific details/explanation of the OBi LED description and behaviour.

LED	Light Pattern	What It Means
Power	Off	The OBi is not receiving power.
	Solid Green	The OBi is operational.
	Flashing Green	The OBi is looking for a DHCP IP address.
	Flashing Orange	The OBi is upgrading. DO NOT remove power!
	Solid Red	The OBi is non-operational.
Status	Off	Reserved for future use
Ethernet	Off	No link detected on the port

	Solid Green	Link detected on the port, but no data transmission activity
	Flashing Green	Active data transmission
FXS n where • $n = 1 - 8$ (OBi508) • $n = 1 - 4$ (OBi504)	Off	Either the port is not enabled or the service corresponding to the PrimaryLine configured for this phone port is not available (but other secondary services may still be available).
	Solid Green	The phone is ready to be used: The service corresponding to the PrimaryLine configured for this phone port is available.
	Flashing green	The phone is in use (the connected phone is off-hook)
	Fast Flashing Green	The phone is ringing (ringing voltage is being applied to the phone port).
	Programmable using the phone port's **MWILedTimer** parameter	New voicemail available (MWI)

LED Pattern When Hardware Reset Button Is Pressed for Ten (10) Seconds:

Power LED blinks green slowly for 5 seconds and fast for 4 seconds preceding unit reboot.

Connecting Accessory Devices to the OBi500 USB Ports*

At present the OBi500 only recognizes OBiBT USB dongles (sold separately by Obihai). You may connect one OBiBT dongle to each of USB Port 1 and USB Port 2. The corresponding services are referred to as OBiBlueTooth 1 and OBiBLueTooth 2 respectively in the device configuration.

*More USB enabled accessories will be supported in the near future with a firmware update.

Internet Connection Set-Up and Configuration

Note that the OBi500 devices can be configured either as a router with Ethernet Port 1 as the WAN Port (to Uplink) and Ethernet Ports 2-4 as 3 LAN switching ports, or as a switch with 4 LAN switching ports. To connect to the network when the device is configured as a switch, connect one end of an Ethernet cable to any one of the four OBi500 Ethernet ports and the other end of the Ethernet cable to an Ethernet port on your Internet router or Ethernet switch. To connect to the network when the device is configured as a router, connect one end of an Ethernet cable to Ethernet Port 1 of the OBi500 and the other end of the Ethernet cable to an Ethernet port on your Internet router or Ethernet switch.

By default the OBi will request an IP, DNS and Internet (WAN) Gateway IP addressing via DHCP.

PHONE Port Set-Up and Configuration

A phone has a very basic UI (User Interface) for I/O (Input / Output) of signalling or control messages.

The OBi PHONE port supports input signalling and control messages comprised of: On Hook, Off Hook, Hook Flash, DTMF tones. Note that OBi5xx only supports touch tone telephones; pulse-dialing is not supported.

The OBi PHONE ports support output signalling and control messages comprised of: Caller ID/CWCID, MWI, DTMF/Tone, Ring, Pol-Rev, CPC, Power Denial.

Each OBi PHONE port has a Maximum Sessions capacity of two (2). This is not configurable.

The OBi PHONE port will reply BUSY to a new incoming call when:

- The PHONE port already has 2 calls in session (or 1 call if call waiting is not enabled on the port)
- The PHONE port is ringing the phone.
- The phone is in a dialing or fast busy ("Invalid" state).
- Already in a FAX call

The OBi PHONE port supports Call Waiting when a 2nd call is an inbound call:

- A Hook Flash (or depressing the Flash button) invokes switching between two (2) calls.
- When the OBi PHONE port goes On Hook this will end current call and invoke a ring for the holding call.

The OBi PHONE port supports 3-way Calling when the second call is an outbound call.

On the first Hook Flash during an active call the OBi can make a second outbound call.

One the second hook flash, the first call and the second outbound call are place in a conference.

To remove the second conferenced party, invoke a third Hook Flash.

When the OBi goes On Hook during a 3-way Call, this will become a transfer when 2nd (outbound) call is ringing or connected. If the 2nd (outbound) call does not succeed, e.g. no answer or busy, then the OBi PHONE port can go to On Hook state and will ring as the holding call is still on the line, or simply hook flash to resume the first call.

The OBi PHONE port can select from the following services with which it can complete a call: SP1 – SP9, OBiTALK (PP1), OBiBlueTooth 1 (BT1) and OBiBlueTooth 2 (BT2) Services.

Using the OBi500 as a Paging System

You can connect the (Audio) Line Out port to an external PA system for paging. The Line Out port is referred to as **ao** (which stands for Audio Out) in call routing rules. With default configuration, a user can broadcast an announcement to the Line Out port by picking up any of the phone ports and dial #0. This is achieved with the rule `{#0:ao}` in the OutboundCallRoute parameter for each phone port. The OBi plays a short beep prior to playing the caller's announcement.

Similarly a remote caller can also be routed to page with a similar routing rule in the Inbound Call Route. For example, adding this rule `{>100:ao}` to **SP1 Service**::**X_InboundCallRoute** parameter will allows a caller at SP1 to send audio out to the Line Out port by calling the extension 100.

Primary Line

You can select the Primary Line for the PHONE *n* Port for *n* = 1, 2, …, and for the AA, respectively, using the parameters **PHONE *n* Port**::**PrimaryLine** and **Auto Attendant**::**PrimaryLine**.

An SP service can be a SIP-based service or Google Voice. In addition, all device models come with the free OBiTALK (peer-to-peer) service. In this document we sometimes refer to any one of these voice services as a *trunk* and a trunk group (TG) is a (comma-separated) ordered list of trunks. If a TG is selected for making an outbound call, the OBi will pick the first available member in that trunk group for the call. Up to four (4) TGs can be defined in an OBi (see the section *Trunk Groups* for detail).

You can make one of the SP Services, OBiBlueTooth 1/2, or TG1/TG2 as the Primary Line for outbound calls. The Primary Line for the PHONE port(s) and the Auto Attendant is configured via the OBi device management web page described herein or the OBiTALK Device Configuration VoIP Service Provider set-up screen also gives the user the option to select a service or TG as the Primary Line. The list below summarizes the choices available for selection as the primary line:

- SP1 Service
- SP2 Service
- SP3 Service
- SP4 Service
- SP5 Service
- SP6 Service
- SP7 Service
- SP8 Service
- SP9 Service
- OBiTALK Service
- OBiBlueTooth 1
- OBiBlueTooth 2
- Trunk Group 1
- Trunk Group 2

When you want to make a call via a service that is not the Primary Line, you will need to dial that service's access code before the destination number.

The default service route access codes are defined as:

- ** 1 : SIP Service Provider 1 or Google Voice Service 1
- ** 2 : SIP Service Provider 2 or Google Voice Service 2
- ** 3 : SIP Service Provider 3 or Google Voice Service 3
- ** 4 : SIP Service Provider 4 or Google Voice Service 4
- ** 81: OBiBlueBooth 1

- ** 82: OBiBlueBooth 2
- ** 9 : OBiTALK Network

Service route access codes for calling from the PHONE port can be customized if necessary by modifying **PHONE** *n* **Port**::**DigitMap** and **PHONE** *n* **Port**::**OutboundCallRoute**. Service route access codes for calling via the Auto Attendant can be customized if necessary by modifying **Auto Attendant**::**DigitMap** and **Auto Attendant**::**OutboundCallRoute**.

Note: Occurrences of `(Mpli)` and `pli` are substituted internally with the corresponding abbreviated trunk name of the selected primary line.

End User Features Available on the OBi

OBiTALK Web Portal:

The OBiTALK Web Portal allows you to manage your OBi endpoints and their relation to other endpoints in your Circle of Trust. OBiTALK is a web portal and OBi configuration utility which helps OBi users configure devices for optimum savings and access applications which make using OBi with even more convenience.

OBi Circles of Trust

The OBiTALK Web Portal is also where you can set-up Circles of Trust. The Circles of Trust provide a means to team-up with other people with OBi devices and endpoints so that everyone's calls can be made as inexpensively as possible.

OBiON iPhone & iPod Touch + Android Smart Phone Apps

After setting up an account and logging in to the OBiTALK portal, users may download applications for their iPhone and Android smart phones.

OBiAPP for PC Soft Phone App

After setting up an account and logging in to the OBiTALK portal, users may download applications for their iPhone and Android smart phones.

OBi Works with Your Existing Services

If you do not want to configure a new service in order to make free calls using your Internet connection you can simply plug in your existing analog line from your telco phone service or connect the line coming from an VoIP service (from an ATA or cable EMTA) to the OBi. Connect your telephone to the OBi's phone port and you are ready to call other users' endpoints on the OBiTALK network.

Bridge Your Services for Optimum Savings & Convenience

With the OBi device, you can bridge multiple services to route calls in the most efficient cost-effective way. You can connect your telco phone service to the OBi at your house and use your PC or Apple iPhone, iPod touch or Android Smart phone to bridge a call from the OBiTALK network to the phone landline or Internet phone service connected to the OBi device at home.

Call Forwarding

Call Forwarding allows you to send incoming calls to another number of your choosing. Calls can be forwarded to a number reachable from the landline service, VoIP service or OBiTALK network. The following types of call forwarding are possible with the OBi:

Call Forward ALL: When you use Call Forward ALL, all calls are immediately forwarded to the number you indicate when you turn on the feature. To enable Call Forward ALL, from a phone attached to the OBi, dial *72. You will be prompted to enter the number to which the calls will be forwarded. Dial the number plus the # key and a confirmation tone will be heard. To disable Call Forward ALL, dial *73. A confirmation tone will be heard.

Call Forward on Busy: When you use Call Forward on Busy, all calls are forwarded to the number you indicate only when you are already engaged in a call with your phone attached to the OBi. To enable Call Forward on Busy, from a phone attached to the OBi, dial *60. You will be prompted to enter the number to which the calls will be forwarded. Dial the number plus the # key and a confirmation tone will be heard. To disable Call Forward on Busy, dial *61. A confirmation tone will be heard

Call forward on No Answer: When you use Call Forward on No Answer, all calls are forwarded to the number you indicate only when you do not answer the call with your phone attached to the OBi. To enable Call Forward on No Answer, from a phone attached to the OBi, dial *62. You will be prompted to enter the number to which the calls will be forwarded. Dial the number plus the # key and a confirmation tone will be heard. To disable Call Forward on No Answer, dial *63. A confirmation tone will be heard.

Caller ID – Name & Number

Caller ID allows you to see the number and (if available) the name of the person calling you. You can use this information to decide whether or not to answer the call. You must have a phone (or device) that supports caller ID to use this feature.

Call Waiting

Call waiting lets you take a second call that comes in when you are already on the phone with another party and not have to disconnect to take the new call. When you are on the line with the first party, you will hear a tone signalling you there is a second call coming in. To answer this call, press the "flash" button on your phone or depress and release the switch hook on the telephone. The first party will be placed on hold and you will be connected to the second party until you press the "flash" button or depress and release the switch hook again.

Since Call Waiting can interfere with fax calls already in progress, it is advised that you configure your fax machine to dial the Cancel Call Waiting code before it dials the destination fax machine.

3-Way Calling

3-Way Calling allows you to talk to two parties at the same time with everyone on a telephone at a different location. To use 3-Way Calling, when you are in a call with another party and want to add a second to the conversation, press the "flash" button or depress and release the switch hook on your phone. You will be presented with a second dial tone and the first party will be placed on hold. Dial the second party. When they answer, you will be able to inform them that you intend to connect them with the first party (now on hold) and have a conference. At this point press the "flash" button or depress and release the switch hook on your phone. This will connect the first party, the second party and yourself. You can all continue to talk together.

Call Transfer (Attended)

You can transfer a call to a third party using the attended transfer capabilities of the OBi. To use Attended Call Transfer, while in a call with the party who will be transferred, press the "flash" button or depress and release the switch hook on your phone. You will be presented with a second dial tone. The party who will be transferred will be placed on hold. Dial the transfer target. When the transfer target answers, you will be able to inform them that you intend to connect them with the party on hold. At this point press the "flash" button or depress and release the switch hook on your phone. This will connect the party to be transferred, the transfer target and yourself. You can continue to talk together, as this is now a 3-way call, or you can hang up the phone and the other two parties will remain connected.

Nordic Region Mid-Call Feature Invocation Style

In the above description of call waiting, 3-way calling, and call transfer operations, the way the features are invoked is referred to as N. America style. In Nordic regions (such as Sweden, Norway), the same features are invoked by hook flashing followed by a digit 0, 1, 2, 3, or 4 to more precisely control which operations to apply to the calls. For these regions, the phones may also be equipped with an R button for hook flashing. The commands issued to the OBi are referred to as R0, R1, R2, R3, R4, and R5. Here is a summary of the operations:

Commands	Operations	Scenarios
R0	Reject the 2nd incoming call	1st call connected, 2nd call ringing
R1	End the 1st call. Resume or answer the 2nd call	1st call connected, 2nd call on hold or ringing
R2	Hold 1st call. Resume or answer the 2nd call (swap calls)	1st call connected, 2nd call on hold or ringing
R3	Keep the 1st call. Resume or answer the 2nd call (conference)	1st call connected, 2nd call on hold or ringing
R4	Transfer 2nd call peer to the 1st call peer	1st call connected, 2nd call on hold or connected

To select the Nordic style of feature invocation, set the parameter PHONE Port::CallCommandSignalMethod to "Nordic Regions (R1, R2, …)". The default is: N. America.

Caller ID Block (Anonymous Calling)*

Caller ID Block allows you to mask your name and number information from appearing on the phone you are calling. To use Caller ID Block for one call only, dial *67 and then the destination number. To use Caller ID Block on a persistent basis, dial *81 from the handset attached to the OBi. All calls will use the Caller ID Block feature until you cancel the Caller ID Block. To cancel Caller ID Block, dial *82 from the handset attached to the OBi.

*This service requires ITSP support. While most ITSP services support this service, Caller ID Blocking is NOT available with Google Voice service at present.

Automatic Call Back (Call Return)

Automatic Call Back, also called Call Return can be used to call back the last caller who called you without actually dialing their number. To use Automatic Call Back, from the phone attached to the OBi, dial *69. The OBi will then attempt to use the previous callers Caller ID information to make the call.

Repeat Dialing

Repeat Dialing is useful when you call a number that is busy and you want to keep trying so that your call gets through when the far end is available. Repeat dialing will continue to try the last number until the OBi device can complete the call or Repeat dialing is cancelled. To enable repeat dialing, from the phone attached to the OBi, dial *05 and hang up. To cancel repeat dialing, from the phone attached to the OBi, dial *06.

Anonymous Call Block

Anonymous Call Block allows you to block calls from incoming callers when there is no identifying caller ID name or number. Incoming calls will be presented with a busy signal. To use Anonymous Call Block, from the phone attached to the OBi, dial *77. To cancel Anonymous Call Block, from the phone attached to the OBi, dial *87.

Do Not Disturb

Do Not Disturb (DND) allows you to set the phone to immediately forward calls made to your OBi to the number set-up as your voicemail number / account. If no voicemail account is set-up, the OBi will return a busy signal to the caller until you

OBIHAI
technology, inc.

turn off DND. To turn on DND, from a phone attached to the OBi, dial *78. To turn off DND, from a phone attached to your OBi, dial *79.

Message Waiting Indication – Visual and Tone Based

Message Waiting Indication allows you to be notified when there is a new voice message for you. The OBi supports both Visual and Tone based Message Waiting Indication. With Tone-based Message Waiting Indication, you will know there is a message for you when you hear a "stutter" dial tone right when you first pick up the phone to make a call. Typically, this stutter tone will be removed once you listen to your message(s). Visual-based Message Waiting Indication will turn on a light or screen icon on your phone (or phone base station) when there is a message waiting for you. Typically, this light or icon will go dark when you have listened to your new message(s).

Speed Dialing of 99 OBi Endpoints or Numbers

The OBi device supports Speed Dialing of 99 numbers. These numbers can be associated with phones reachable via an Internet or landline service or the OBiTALK network. Be careful with the Speed Dial Set-Up as this will conflict with the Speed Dials set-up on the OBiTALK portal. The Speed Dials that are set-up on the OBiTALK portal will always overwrite anything set-up via the phone connected to the OBi.

PHONE-to-PHONE Collaborative Features

While each phone port functions independently of each other, an OBi500 device provides PBX-like features that tie the ports together to work as a phone system. The default the phone port digit map and call routing rules are set up such that you can:

- Dial **#n** to call phone port *n*, *n* = 1, 2, 3, … 8*

- Dial **#*** to ring all the phones at once

- Dial **#0** to page out of the Line Out Port

*Dialing #n from phone *n* is reserved for checking voicemail in phone *n*'s mailbox

Furthermore, when calling phone-to-phone with a prefix *96, the caller can trigger the following behaviour:

- If the called phone is idle (on-hook), it will ring normally with a special Caller-ID that indicates the call is from the other PHONE Port

- If the called phone is on-hook with a call on-hold, the calling phone will pick up and resume that call

- If the called phone is ringing, the calling phone will pick up and answer that call

- For all other scenarios, the calling phone will hear busy tone

You can also transfer an external call from PHONE A to PHONE B the usual way: while connected on an external all, hook flash and dial #*n* extension to ring the other phone, then hang up to transfer when the caller phone rings or answers.

For incoming calls on any trunk, one can set up the corresponding inbound call route to ring any combination of the phone ports. The default inbound call routes are setup to ring just Phone 1 and Phone 2

For outgoing calls, each phone port has its own digit map and outbound call route configuration, which means that you have the full flexibility in allocating trunks for making calls from each port independently. Each port may also have a different primary line assigned; the default, however is to set the primary line to SP1 for all phone ports.

Star Code Features

The OBi device supports service features via the handset connected to the PHONE port. The following Star Codes can be used to access the indicated features. OBi Star Code Enabled Features Apply to All Voice Services.

*03, Request peer device to loopback media in the next outbound call

*04, Request peer device to loopback RTP packets in the next outbound call

*05, Tell device to periodically redial the last called number until the called party rings or answers

*06, Cancel the last repeat dial request

*07 Redial

*69 Call Return

*81 Block Caller ID (Persistent Mode)

*82 Unblock Caller ID (Persistent Mode)

*67 Block Caller ID (One Time)

*68 Unblock Caller ID (One Time)

*72 Call Forward All (Enter Number + #)

*73 Disable Call Forward All

*60 Call Forward on Busy (Enter Number + #)

*61 Disable Call Forward in Busy

*62 Call Forward on No Answer (Enter Number + #)

*63 Disable Call Forward No Answer

*77 Block Anonymous Calls

*87 Unblock Anonymous Calls

*56 Enable Call Waiting

*57 Disable Call Waiting

*78 Do Not Disturb – Turn On

*79 Do Not Disturb – Disable

*74 Speed Dial Set-Up (Enter SD No. [1-99] then Tel No. + #) ∞

*75 Speed Dial Read-Back (Enter SD No.)

*76, Clear A Speed Dial

*96, Barge In

*98, Blind Transfer

*4711, Use G711 Only on the next outbound call

*4729, Use G729 Only on the next outbound call

*28, Make OBiBlueTooth 1 discoverable for 2 minutes

*29, Make OBiBlueTooth 2 discoverable for 2 minutes

∞ Note: Be careful with the Speed Dial Set-Up as this will conflict with the Speed Dials set-up on the OBiTALK portal. The Speed Dials that are set-up on the OBiTALK portal will always overwrite anything set-up via the phone connected to the OBi.

Call Forward Numbers

There is one set of Call Forward Settings per voice service on the OBi, such that the settings apply to incoming calls on that service only. However calls may be forwarded to numbers on the same service or on another service. Therefore each call forward number stored in the OBi configuration MUST include call routing information to let the device know which voice service should be used to forward the call to. The general format of a call forward number is:

TK(*number*)

where *TK* is the abbreviated name of a voice service. Valid values of *TK* are `SP`*n* for the SP*n* Voice Service where *n* = 1 – 9, `BT1`, `BT2` for OBiBlueTooth 1/2, or `PP` for the OBiTALK Service.

The *number* to forward to must be in the final form that is acceptable by the service provider. OBi will not apply any Digit Map or Call Routing Rules on it.

Examples: `SP1(14089991234)`, `PP1(ob200333456)`

You may also set the call forward number to PH1 – PH8 (a phone port), AA (the auto attendant), PK (the call park), AI (Audio Line In), or AO (Audio Line Out).

Configuration and Management Interfaces of the OBi Device

Telephone-IVR-Based Local Configuration

The OBi utilizes an interactive voice response (IVR) system for both its configuration and day-to-day function. The IVR is, in essence and automated attendant the OBi user will access to either invoke a verbal response from the OBi to provide information to the user (such as IP address) or instruct the OBi to act on the routing / placement of a call to a particular interface. More information about the Auto Attendant IVR for OBi call processing will be provided later in the document.

There are two IVR menus.

1. Auto Attendant IVR 1: Referred to as **aa** (synonymous to **aa1**) for call processing commands. Unless stated otherwise, the Auto Attendant operations discussed in this document refers to this AA instance
2. Auto Attendant IVR 2: Referred to as **aa2** for local configuration. Unless stated others, the IVR operations discussed in this document refers to this AA instance

If one or more the changes to the settings require a reboot to take effect, it will be done automatically when quitting the IVR. As a default, the IVR (AA2) can be invoked by dialing *** from any phone port.

Tip: By pressing the appropriate button sequence on the telephone key pad, you can barge into the next menu of the IVR or invoke a command without first waiting for the previous announcement to end.

IVR Main Menu

Main Menu configuration options are accessed by pressing * * * from a phone attached to the PHONE port of the OBi, followed by a single digit of the option number as listed below:

Selection	Announcement	What Can You Do?
1	Basic Network Status Your IP address and DHCP status will be read back to you.	Press 0 to repeat the information.
2	Advanced Network Status Your primary & back-up DNS server, primary & back-up NTP server will be read back to you.	Press 0 to repeat the information.
3	DHCP Current Value Your current value will be read back to you and you will be given the option to change the value	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information.
4	IP Address Current Value Your current value will be read back to you and you will be given the option to change the value. If you elect to enter a new value (static IP address) DHCP	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information.

	will be disabled.	
5	Password Current Value Your current IVR password value will be read back to you and you will be given the option to change the value.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information.
6	Please Wait (while OBi is checking for software update)… This is followed by either: - Software Update Available. Press 1 to update software, OR - Software Update Not Available	If an update is available, press 1 to proceed with the update. The software update process will start as soon as you hang up the phone. Warning: Once the software upgrade process starts, the device's power LED will blink rapidly. Please make sure the power and network cable stay connected to the unit until the process is complete.
8	Restore Factory Default	Press 1 to confirm device restore to factory default settings. Press # to return to device configuration menu. Press # # to exit IVR.
9	Reboot OBi Device	Press 1 to confirm device reboot. Press # to return to device configuration menu. Press # # or hang up to exit IVR.
0	Additional Options Access other configuration options of the OBi device.	Enter option followed by the # key.

System Level Configuration Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0**:

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
1	Firmware Version The current value of the firmware version will be read back.	Press 0 to repeat the information. Press # to enter another configuration selection.
2	IVR Password	Press 1 to enter a new value.

	The current value of the IVR password will be read back.	Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
3	Debug Level The current value of the debug level will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
4	Syslog Server IP Address The current IP address of the syslog server will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
5	Syslog Server Port The current value of the syslog server port will be read back.	Press 1 to enter a new value. Press 2 to set the default value of 514. Press 0 to repeat the information. Press # to enter another configuration selection.

Network Related Configuration Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0**:

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
20	DHCP Configuration The current value of the DHCP configuration will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
21	IP Address The current value of the IP address will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

		Press 1 to enter a new value.
22	Default Gateway The current value of the default internet gateway will be read back.	Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
23	Subnet Mask The current value of the subnet mask will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
24	DNS Server (Primary) The current value of the primary DNS server will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
26	NTP Server (Primary) The current value of the primary NTP server will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

SIP Service Provider Configuration Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0** for SIP Service Provider One (SP1):

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
100	Enable Service Provider One (SP1) The current value will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
101	Registration State of SP1 The current value will be read back.	Press 0 to repeat the information. Press # to enter another configuration selection.
102	SP1 User ID	Press 1 to enter a new value.

	The current value will be read back.	Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
167	SP1 Block Caller ID Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
168	SP1 Block Anonymous Call Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
172	SP1 Call Forward ALL – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
173	SP1 Call Forward ALL Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
174	SP1 Call Forward on Busy – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
175	SP1 Call Forward on Busy Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
176	SP1 Call Forward on No Answer –	Press 1 to enter a new value.

	Enable / Disable	Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
177	SP1 Call Forward on No Answer Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0** for SIP Service Provider Two (SP2):

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
200	Enable Service Provider One (SP2) The current value will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
201	Registration State of SP2 The current value will be read back.	Press 0 to repeat the information. Press # to enter another configuration selection.
202	SP2 User ID The current value will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
267	SP2 Block Caller ID Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
268	SP2 Block Anonymous Call Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information.

		Press # to enter another configuration selection.
272	SP2 Call Forward ALL – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
273	SP2 Call Forward ALL Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
274	SP2 Call Forward on Busy – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
275	SP2 Call Forward on Busy Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
276	SP2 Call Forward on No Answer – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
277	SP2 Call Forward on No Answer Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

OBiTALK Configuration Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0**:

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
900	Enable OBiTALK Service The current value will be read back.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
901	Registration State of OBiTALK The current value will be read back.	Press 0 to repeat the information. Press # to enter another configuration selection.
967	OBiTALK Block Caller ID Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
968	OBiTALK Block Anonymous Call Enable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
972	OBiTALK Call Forward ALL – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
973	OBiTALK Call Forward ALL Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
974	OBiTALK Call Forward on Busy – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value.

		Press 0 to repeat the information.
		Press # to enter another configuration selection.
975	OBiTALK Call Forward on Busy Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
976	OBiTALK Call Forward on No Answer – Enable / Disable	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.
977	OBiTALK Call Forward on No Answer Number	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

Auto Attendant Configuration Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0**:

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
80	Enable / Disable Auto Attendant.	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection.

Customized AA Prompt Recording Options

Additional Configuration Options Available with the OBi IVR after Pressing *** * * 0**:

Selection (Always Press "#" After Entering Selection)	Announcement	What Can You Do?
1001	Option 1001 current value is: (the recorded prompt)	Press 1 to enter a new value. Press 2 to set the default value. Press 0 to repeat the information. Press # to enter another configuration selection. Note: After pressing 1 to record a new prompt, the OBi says "Enter value followed by the # key". At that point, you can press any digit (0-9) to start recording, and then press # to end recording. Tips: Leave about 1s of gap at the end of recording to avoid unintended truncation by the OBi. After a new prompt is recorded, Obi immediately plays back the recorded audio, and then presents the following options: Press 1 to save (save the recorded prompt permanently in long term memory) Press 2 to re-enter (the last recorded prompt is discarded) Press 3 to review Press # to cancel (the last recorded prompt is discarded)
Similarly for Options 1002 - 1020		

With these options you can record up to 10 prompts which can be arranged in any combinations and used as customized AA prompts. Each prompt recording is limited to 60s, where the prompt duration is rounded to the nearest number of seconds. A total of 122s is available to store all the recordings. The device will reboot automatically when you hangup if any of the prompts have been modified and saved. Furthermore you can enter a text description for each recorded prompt as a reminder of the contents of the prompt (under the Voice Services - Auto Attendant configuration page).

Remote Provisioning

This is the process by which OBi devices download a configuration file from a server, which may be located in the cloud or in the same enterprise. The configuration file may contain all the necessary parameter values for the device to function normally, it may also tell the device to download an additional configuration file from a different URL, or to download a different firmware to replace the current one, and so on. The configuration file format and parameter naming conventions are proprietary to Obihai but are common across all Obihai products.

There are currently two configuration file formats supported: A full XML format with the XML tags in full text and a short XML format with the XML tags substituted with a single letter abbreviation. The XML structure and parameter naming convention closely follows TR-069/TR-104. For a full description of the configuration file and parameter names, please refer to the *OBi Device Provisioning Guide*.

Similar to the way parameters are grouped under different device configuration web pages, parameters are grouped into a number of configuration objects for remote provisioning. In fact you will find a near one-to-one correspondence between these objects and their location within the configuration web pages. To illustrate this, consider the web page SP1 Service, the SIP Credentials section:

The corresponding configuration object in a phone configuration XML file is:

VoiceService.1.VoiceProfile.1.Line.1.SIP.

as shown below:

```
<Object>
  <Name>VoiceService.1.VoiceProfile.1.Line.1.SIP.</Name>
  <ParameterValueStruct>
    <Name>AuthUserName</Name>
    <Value>john.j.smith@gmail.com</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>AuthPassword</Name>
    <Value>zYz123#$12</Value>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>URI</Name>
    <Value X_UseDefault="Yes"/>
  </ParameterValueStruct>
  <ParameterValueStruct>
    <Name>X_MyExtension</Name>
    <Value>16188</Value>
  </ParameterValueStruct>
```

```xml
    <ParameterValueStruct>
      <Name>X_XsiUserName</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>X_XsiPassword</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>X_XmppDomain</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>X_XmppUserName</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>X_ContactUserID</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
    <ParameterValueStruct>
      <Name>X_EnforceRequestUserID</Name>
      <Value X_UseDefault="Yes"/>
    </ParameterValueStruct>
  </Object>
```

Note that the dot (.) at the end of the object name is part of the name that must not be omitted in the XML file. You must use the correct object name in order to create a valid configuration file for the phone. You can find the object name corresponding to each configuration web page/section listed at the end of this document.

The list of configuration parameters, syntaxes, and valid values for a specific device mode are collectively known as a *data model* of the device. The most up-to-date data model for the OBi500 can be found online at the following URL:

http://www1.obitalk.com/Downloads/dev/datamodel/obihai504.xml

http://www1.obitalk.com/Downloads/dev/datamodel/obihai508.xml

ZT (Zero Touch): Device Customization at Obihai's Factory

When products are shipped from the factory, they come with a set of default parameter values installed by Obihai for all customers. A service is also available from Obihai such that products shipped to a particular customer can have a small number of parameter values customized for that customer. For example, a very useful parameter to customize is the **ITSP Provisioning**::**ConfigURL** parameter which tells the phone where to download a configuration file. With this, the first time a new phone is powered on and connected to the network, it can automatically contact the designated URL to get the initial configuration file; hence the name "Zero-Touch".

Note: ZT devices must contact OBiTALK.com one time to get the customized values before they can start normal operation. Make sure the device can access the Internet before first use.

Web Server-Based Local Configuration

The OBi device has an integrated device management web server which can be accessed from a PC or similar device using a browser. Although all popular browsers are tested for compatibility with the OBi device management web server, there may be inconsistencies which arise from time to time. Please contact support@obihai.com if you have any questions about the OBi device management web server and how it appears in your browser window.

Access the OBi Device Management Web Page:

1. From a phone attached to the OBi, dial *** * *** to access the OBi Config Attendant.
2. Choose "**1**" to hear the IP Address of the OBi read back to you. Write this down.
3. Enter the OBi IP Address in a local PC web browser address field:
4. When prompted, enter "admin" for user name and "admin" for password.

When you access the OBi device management web page, you will be prompted for a user name and password. There are two levels of access to the OBi web page – User Level and Admin Level. The default "user name / password" for User Level access is "user / user". The default "user name / password" for Admin Level access is "admin / admin". The Admin and/or User passwords may have been changed using the OBi device web page, provisioning by a service provider or via the OBiTALK web portal (Admin only). Please be sure you have access to the correct Admin or User password before you attempt to log on to the OBi Device Management Web Page.

The OBi device management web page is organized into sections to allow for a manageable and compartmentalized approach to configuring the many hundreds of parameters available on the OBi device. Use the expandable / collapsible menu tree on the left side of the page to easily navigate the various configuration parameter sections of the OBi device.

IMPORTANT: Every configuration page must be submitted individually after changes made on the page. Otherwise those changes will be discarded once you navigate to another page. Most changes will require a reboot of the unit (by clicking the reboot button for instance) to take effect. However, you may reboot the unit just once after you have made and submitted all the necessary changes on all the pages.

When the device is operating in router mode (OBi202 only), the built-in web server can be accessed from the LAN side or the WAN side. While access from the LAN side is always allowed, the access from the WAN side may be disabled by configuration for security reasons. In fact the WAN side access to the web server is disabled by default. You can enable this option on the device web page (from the LAN side), or by using the device configuration IVR (* * * 0 option 30).

Web Page Conventions and Icons & Buttons:

Icon / Button	Description	Remark
	This icon indicates that there is more information available which might describe the workings, limits or thresholds for the parameter to which it is adjacent. You can mouse over this icon to reveal this information.	
	When a modification has been made to a parameter on a page, the Submit button MUST be clicked before proceeding to another page.	

	If you make changes to a parameter on a page and you do not want to keep them for submission, click the "Clear Changes" button to revert back to the parameter setting present before the most recent change was entered.	
	Click the "Use Defaults Only" button if you want to revert all parameters on a given page to their Default settings. If you want to revert just one or two parameters on a page to default settings you should use the Default check box found on the right side of the parameter. See next Item.	You will be prompted to confirm that you want all the parameters on the page to revert back to system default settings.
Default ☑	When you wish to modify a parameter away from its default setting, you should un-check the 'Default' box. This will open the parameter field for access and modification. If there is a non-default setting in a parameter field and you want to revert that parameter back to its default setting, check the "Default" box and the default setting will appear.	Default value of a parameter may be changed with a firmware upgrade. Leaving a parameter at default setting allows the device to use proper default value with the firmware currently installed in the device
	This icon indicates that the configuration currently programmed on the OBi device is "set" and "running". No reboot is necessary if you have submitted configuration modifications.	This icon does not indicate the currently running configuration is working properly.
	After Submitting changes to a web page on the OBi, the "Reboot Required" icon may appear. In order for the modifications to run, you will need to reboot the OBi.	You can continue to make modifications to OBi parameters – on separate pages if necessary – before you reboot and "set" the modifications in the running system.
Reboot	The "Reboot" button is used when the "Reboot Required" icon appears indicating the OBi device requires a reboot to invoke one or more parameter modifications.	When performing a System Configuration Reset, the Reboot button does not need to be pressed. The OBi will reboot automatically when the "Reset" button is selected.
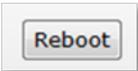
Device Web Pages Listing

The local device web page covers all the configurable options. In the sections that follow we will describe each of the configuration options by going through all the device web pages. The available pages are listed in a left-side-panel of each web page for random access. Here is the layout of the side panel where each bullet represents a separate configuration web page:

Status
- System Status
- LAN Status
- Call Status
- Call History
- SP Services Stats
- PHONE Port Status
- Call Park Status

Router Configuration
- WAN Settings
- LAN Settings
- DHCP Reservation
- Firewall and DMZ
- Port Forwarding
- QoS Settings

System Management
- Auto Provisioning
- Device Admin
- Device Update

Service Providers
ITSP Profile A (*Repeated for ITSP Profile B, C, …, H and I*)
- General
- SIP
- RTP

Voice Services
- SP1 Service
- SP2 Service
- SP3 Service
- SP4 Service
- SP5 Service
- SP6 Service
- SP7 Service
- SP8 Service
- SP9 Service
- OBiTALK Service
- Auto Attendant
- Gateways and Trunk Groups
- OBiBlueTooth 1
- OBiBlueTooth 2
- Hunt Groups

Physical Interfaces

OBIHAI
technology, inc.

- Line Out Port
- Line In Port
- PHONE1 Port
- PHONE2 Port
- PHONE3 Port
- PHONE4 Port
- PHONE5 Port
- PHONE6 Port
- PHONE7 Port
- PHONE8 Port

Codec Profiles
- Codec Profile A
- Codec Profile B

Tone Settings
- Tone Profile A
- Tone Profile B

Ring Settings
- Ring Profile A
- Ring Profile B

Star Codes
- Star Code Profile A
- Star Code Profile B
- Star Code Profile C
- Star Code Profile D
- Star Code Profile E
- Star Code Profile F
- Star Code Profile G
- Star Code Profile H

User Settings
- Speed Dials
- User Defined Digit Maps

Status Web Pages

System Status Web Page

This page shows the service status on the device. The table below describes all the status parameters reported on this web page.

Status Parameter	Description	Example Value
WAN Status		
AddressingType	Method currently used by the device to get an IP address assignment	DHCP
IPAddress	IP address currently assigned to the device	192.168.15.165
SubnetMask		255.255.255.0
DefaultGateway		192.168.15.1
DNSServer1		4.2.2.2
DNSServer2		
MACAddress	MAC address installed on the device	9CADEF90004E
LLDP-MEDStatus		
LTE STatus		
AddressingType		
IPAddress		
SubnetMask		
DefaultGateway		
DNSServer1		
DNSServer2		
MACAddress		
Product Information		
ModelName		OBi508
MACAddress		9CADEF90004E
SerialNumber		88H01NA00ZXV
OBiNumber		552 860 300
HardwareVersion		1.1
SoftwareVersion		4.0.1(Build: 4256)
SystemTime	Shows the current time on the system	15:32:35 01/29/2014, Wednesday
UpTime	With last reboot reason in parathesise. See the list below this table for a list of reboot reason codes	20 Days 5:04:13 (2)
CertificateStatus	Indicate if a device certificate is installed on the unit	Installed
CustomizationStatus	Indicate if this device is a customized unit	Generic
OBiBT Dongle n Status (repeated for n = 1, 2)		
Status		
Discoverable		
CallState		0 Active Calls
BindingService		
SPn Service Status (repeated for n = 1 – 9)		
Status	Registration status of this service. If there are problems with the registration or authentication, the SIP 4xx – 6xx error code and error message will be displayed here. This is very useful information for	Registered (server=192.168.15.118; expire in 39s)

	troubleshooting issues with SIP-based services.	
PrimaryProxyServer	IP address of the current Primary Proxy Server if proxy server redundancy is enabled on this service	
SecondaryProxyServer	IP address of the current Secondary Proxy Server if proxy server redundancy and secondary registration are both enabled on this service	
CallState		0 Active Calls
OBiTALK Service Status		
Status	Connection status with the OBiTALK network	Normal (User Mode)
CallState		0 Active Calls

Reboot Reason Codes

0: Reboot on Power Cycle
1: Operating System Reboot
2: Reboot after Firmware Update via provisioning or phone (***6)
3: Reboot after New Profile Invoked
4: Reboot after Parameter Value Change or Firmware has changed and invoked via device web page
5: Reboot after Factory Reset using the OBi device hardware pin
6: New Profile Invoked AND Profile URL Changed
7: Reboot from SIP Notify (Reserved)
8: Reboot from Telephone Port (IVR)
9: Reboot from Webpage - No change in parameter value(s) or firmware
10: Reboot During OBiTALK Signup
11: Reboot During OBiTALK Signup
12: Reboot after DHCP server offers IP, GW-IP and/or Netmask different from what the OBi device is currently using
13: Reboot on Data Networking Link Re-establishment

Call Status Web Page

The Call Status page shows a number of running call statistics and state parameters for each active call currently in progress. The following information and statistics are shown for each call:

Status	Description
Peer Name	Call Peer's Name
Peer Number	Call Peer's Number
Start Time	Starting time of the call
Duration	Duration of the call
Peer RTP Address	The peer address:port where RTP packets are sent to
Local RTP Address	The local address and port where RTP packets are sent from
RTP Transport	The transport used for RTP (UDP, TCP, or SSL)
Audio Codec	The audio encoder and decoder being used for this call
RTP Packetization	The transmitted and received packet sizes in milliseconds
RTP Packet Count	Total number of RTP packets transmitted and received thus far
RTP Byte Count	Total number of RTP bytes transmitted and received thus far
Peer Clock Differential Rate	Clock difference between this device and the peer in ppm (parts per million)
Packets Out-of-Order	Number of received RTP packets that are out of order

Packets Lost	Number of incoming RTP packets assumed lost
Packet Loss Rate	Amount of incoming RTP packets assumed lost rate in percent
Packet Drop Rate	Amount of incoming RTP packets dropped in percent
Jitter Buffer Length	Size of the current jitter buffer in milliseconds
Received Interarrival Jitter	Average measured network jitter in the received direction in milliseconds
Max Interarrival Jitter	Maximum measured network jitter in the received direction in milliseconds
Jitter Buffer Underruns	Amount of jitter buffer underruns during the call

For each entry on the call status page, the following buttons may be available:

- **Remove**: This button is available for all calls. Pressing this button will end that call.

- **Record**: This button is available for calls involving the Phone port only. Pressing this button allows you to record the current conversation in an audio (.au) file

Call History Web Page

The OBi Call History page shows the last 400 calls. Detailed call information is available, including what terminals were involved, the name (if available) of the Peer endpoints making the call and the direction / path the call took.

The Call History page also captures what time various events took place.

The Call History can be saved at any time by clicking on the "Save All" button. The Call History can be saved as an XML formatted file called "callhistory.xml".

SP Services Stats Web Page

This page shows the following information for each SP*n* service, where *n* = 1 – 9. There is only one writable setting is this group: **ResetStatistics**. The rest of the parameters are read-only status.

Parameter	Description	Example Value
Reset Statistics		
ResetStatistics	This is a Boolean option to reset the statistics for this SP Service. After submitting this change, the value is automatically reverted back to the default value `false`.	`false`
RTP Statistics		
PacketsSent	Total RTP packets sent on this line	1234
PacketsReceived	Total RTP packets received on this line	1045
BytesSent	RTP payload bytes sent for this line	1023480
BytesReceived	RTP payload bytes received for this line	1345320
PacketsLost	Number of RTP packets lost on this line	2
Overruns	Number of times receive jitter buffer overrun on this line	0
Underruns	Number of times receive jitter buffer underrun on this line	0

PHONE Port Status Web Page

This page shows the following information for each Phone Port.

Parameter	Description	Example Value
Port Status		
State	Port status, such as on-hook, off-hook, ringing	On Hook
LoopCurrent	Loop current in mA	0 mA
VBAT	PHONE port battery voltage in volts. Not applicable for LINE port	58 V
TipRingVoltage	Sensed differential Tip/Ring voltage in volts	45 V
LastCallerInfo	Caller ID of previous call	--

Call Park Status Web Page

This page shows the following information for each parked call.

Parameter	Description	Example Value
Call Park Status		
"Remove" (Checkbox)	Check this option and press the "Remove" button to remove this call from the parking lot by ending the call	
"Send To" (Number)	Enter a destination number such as SP1(14085551234) and press the "Send To" button to transfer the parked caller to that number	
Slot ID	The 2-digit parking slot ID that the call is parked at	
Caller-ID	The caller's name and number, if available	
Time	The time at which the call is parked	
Duration	How long the call has been parked	
Trunk	Which trunk the call is coming from	

Router Configuration Web Pages

The four Ethernet ports on the OBi500 by default work as a 4-port switch, but the device can also be configured to work as a router by setting the parameter **LAN Settings:: OperationMode** = `Router`. When working as a router, there are two possible WAN interfaces (to connect to the uplink). The primary interface is the Ethernet Port 1; the secondary interface is a LTE USB adapter connected to one of the USB ports, to connect to a 4G/LTE network for Internet access. The secondary interface will be used only when the primary interface is not available; the device is said to have *failed over* to the secondary interface. The device will *fail back* to the primary interface when it becomes available again. The device configuration parameters that control the failover/failback operations are grouped under the parameter group **WAN Failover Control**.

WAN Settings Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
Internet Settings (DeviceInfo.WAN.)				
AddressingType	The method used for assigning IP address, subnet mask, default gateway, etc., to the device. Available choices are: • `DHCP`: IP address, default gateway, etc. are assigned by DHCP Server • `Static`: IP address, default gateway, etc. are taken from the manually configured values • `PPPoE`: IP address default gateway, etc. are acquired by PPPoE Protocol	Choice	RW	`DHCP`
IPAddress	The IPv4 address to assign to the device when **AddressingType** is set to `Static`	String	RW	
SubnetMask	The subnet mask to use when **AddressingType** is set to `Static`	String	RW	
DefaultGateway	The default gateway IP address to assign to the device when **AddressingType** is set to `Static`	String	RW	
DNSServer1	IP address of the first DNS server to use, in addition to the ones obtained from the DHCP server when DHCP is also enabled. If AddressingType is set to Static, the device only uses DNSServer1 and DNSServer2 for DNS lookup. It will try up to 5 DNS servers when attempting to resolve a domain name. DNSServer1 and DNSServer2 will be tried first, whichever is specified, and then the ones obtained from the DHCP Server if available	String	RW	
DNSServer2	IP address of the second DNS server to use, in addition to the ones obtained from the DHCP server when DHCP is also enabled. If AddressingType is set to Static, the device only uses DNSServer1 and DNSServer2 for DNS lookup. It will try up to 5 DNS servers when attempting to resolve a domain name. DNSServer1 and DNSServer2 will be tried first, whichever is specified, and then the ones obtained from the DHCP Server if available	String	RW	
MTUSize	MTU size in number of bytes	UINT	RW	`1500`
MACAddressClone	The MAC Address to clone (instead of using the factory	String	RW	

	installed MAC Address)			
PPPoEACName	PPPoE access concentrator name. Enter if it is required	String	RW	
PPPoEServiceName	PPPoE service name. Enter if it is required	String	RW	
PPPoEUsername	PPPoE account username provided by your ISP	String	RW	
PPPoEPassword	PPPoE account password	String	RW	
PPPoEKeepAlive	PPPoE keep alive period in seconds	UINT	RW	60
VLANEnable	Enable VLAN Operation	Bool	RW	false
VLANID	Valid range is 0 – 4094 (4095 is reserved). 0 means VLAN is disabled and egress packets are not tagged by the device. This setting applies to all packets sent by the device	UINT	RW	0
VLANPriority	Valid choices are 0 – 7. This setting applies to all packets sent by the device.	UINT	RW	0
LLDP-MED	Enable LLDP-MED Discovery	Bool	RW	false
LLDP-MEDExclusivePeriod	Delay in seconds before getting or setting up an IP address based on AddressingType to exclusively perform LLDP-MED	UINT	RW	5
LLDP-MEDAssetID	The LLDP-MED Asset ID to broadcast during LLDP-MED Discovery	String	RW	$OBN
Local Time				
CurrentLocalTime	Current local date and time of the device	String	R	
Time Service Settings (**DeviceInfo.Time.**)				
NTPServer1	Hostname or IP address of the first NTP server	String	RW	pool.ntp.org
NTPServer2	Hostname or IP address of the second NTP server	String	RW	

LocalTimeZone	Local time zone. Available choices are: • `GMT-12:00(Int'l Dateline West)` • `GMT-11:00(Samoa)` • `GMT-10:00(Hawaii)` • `GMT-09:00(Alaska)` • `GMT-08:00(Pacific Time)` • `GMT-07:00(Mountain Time)` • `GMT-06:00(Central Time)` • `GMT-05:00(Eastern Time)` • `GMT-04:30(Caracas)` • `GMT-04:00(Atlantic Time)` • `GMT-03:30(Newfoundland)` • `GMT-03:00(Buenos Aires,Greenland)` • `GMT-02:00(Mid-Atlantic)` • `GMT-01:00` • `GMT+00:00(London,Lisbon)` • `GMT+01:00(Rome,Paris,Madrid)` • `GMT+02:00(Athens,Cairo)` • `GMT+03:00(Moscow,Baghdad)` • `GMT+04:00(Abu Dhabi)` • `GMT+04:30(Kabul)` • `GMT+05:00(Islamabad,Karachi)` • `GMT+05:30(New Delhi)` • `GMT+05:45(Kathmandu)` • `GMT+06:00` • `GMT+07:00(Bangkok,Jakarta)` • `GMT+08:00(Beijing,HK,Singapore)` • `GMT+09:00(Tokyo,Seoul)` • `GMT+09:30(Adelaide)` • `GMT+10:00(Sydney,Guam)` • `GMT+11:00(Solomon Is.)` • `GMT+12:00(Fiji,Auckland)` • `GMT+13:00`	Choice	RW	`GMT-08:00(Pacific Time)`
DaylightSavingTimeEnable	Enable daylight saving time on the unit	Bool	RW	`true`
DaylightSavingTimeStart	Daylight Saving Time Start Date. Format: month/day/weekday/hh:mm:ss, where month=1-12, day=±(1-31), weekday=0,1-7 (0=special, 1=Monday, 7=Sunday), hh=0-23,mm=0-59,ss=0-59. If weekday=0, daylight saving starts on the given month/day; otherwise it starts on the weekday on or after the given month/day if day > 0, or on the weekday on or before the last-day-of-given-month+day+1 (note that day = -1 equivalent to last day of the month). :ss may be omitted if the value is 0; :mm:ss may be omitted if mm and ss are both 0.	String	RW	`3/8/7/2`
DaylightSavingTimeEnd	Daylight Saving Time End Date. Same format as Start Date	String	RW	`11/1/7/2`

DaylightSavingTimeDif f	Amount of time to add to current time during Daylight Saving Time. Format: [-]hh:mm:ss. :ss may be omitted if it is 0; :mm:ss may be omitted if both are 0.	String	RW	1
WAN Failover Control (**DeviceInfo.WAN.FailoverControl.**)				
Mode	Method to use to drive the failover from Ethernet to LTE. Choices are: • Disable • Ping • DNS Query	Choice	RW	Disable
Targets	A comma separated list of servers to use to test for failover	String	RW	8.8.8.8, 8.8.4.4, 4.2.2.2, 209.244.0.3
Interval	Failover testing interval in seconds.	UINT	RW	30
Threshold	Number of consecutive negative test results to trigger failover.	UINT	RW	3
PrimaryFailoverDelay	Delay in seconds before restarting test for failover after failing back to the primary interface (Ethernet)	UINT	RW	300
SecondaryFailoverDela y	Delay in seconds before restarting test for failing back after failover to the secondary interface (LTE)	UINT	RW	300
KeepInterfaceAwakeP eriod	Interval to send keep alive traffic on the LTE interface in order to keep it awake. Setting the value to 0 to disable this keep alive feature.	UINT	RW	60
DNS Control (**X_DNSControl.**)				
DNSQueryOrder	Controls the order in querying DNS Servers. The following choices are available: • DNS Server1, DNS Server 2, DHCP Offered DNS Servers • DHCP Offered DNS Servers, DNS Server1, DNS Server 2 • DNS Server1, DNS Server 2	Choice	RW	DNS Server1, DNS Server 2, DHCP Offered DNS Servers
DNSQueryDelay	Controls the delay (in seconds) before trying the next DNS server when resolving a domain name. Available choices are: 0, 1, 2, 3, 4, 5	Choice	RW	2
Local DNS Records (**X_LocalDNSRec.**)				
N	One of 32 Local DNS Records (numbered 1 – 32). Each	String	RW	

where N = 1 – 32	record is a mini script of the following format:			
	Name=A,A,A,... **OR**			
	Name=R,R,R,...			
	where *Name* represents the domain name to be resolved locally, and has the format *prefix+domain* (such as `machine.sip+obihai.com`). Everything after '+' is considered as the *domain* to be appended to the *host* field in each *R* on the right hand side. '+' is optional; if missing the full domain must be used in every *R*.			
	A represents an A record which is just an IP address, such as 192.168.12.17.			
	R represents an SRV record and has the format: {*host*:*port*,*pri*,*wt*} where			
	- *host* is a hostname with or without domain part (such as xyz, xyz.abc.com.). A dot (.) at the end of *host* indicates it is a complete hostname that does not require the domain to be appended.			
	- *port* is a port number (such as 5060)			
	- *pri* is the priority. Valid value is 0(highest) – 65535(lowest)			
	- *wt* is the weight. Valid value is 0(lowest) – 65535(highest)			
	wt is optional; 1 is the default if not specified.			
	pri is optional only if *wt* is not specified; 1 is the default if not specified.			
	port is optional; the default to use will be based on the protocol (5060 for SIP, 80 for HTTP, etc.) .			
	The enclosing curly braces { } are also optional if there is only one *R*; or			
	if there is no comma used inside the *R*.			
	Examples:			
	`_sip._udp+obihai.com=abc,xyz,pqr:5080,{mmm,2},{super.abc.com.}`			
	`abc.obihai.com=192.168.15.118,192.168.15.108`			
	Note: If the A record of a given hostname cannot be found in any of the Local DNS Records, the device will attempt to resolve it using external DNS queries. Any change applied to local DNS Record needs reboot in order to take effect.			

LAN Settings Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
LAN Settings (**DeviceInfo.LAN.**)				
CurrentRouterIPAddress	Current LAN side address of the router	String	R	
OperationMode	Choices are:	Choice	RW	`Bridge`

OBIHAI
technology, inc.

	• Router • Bridge			
RouterIPAddress	The IP address of the router on the LAN side. If it conflicts with its WAN side IP address, the device will adjust it automatically to resolve the conflict. This parameter is valid in router mode only	String	RW	192.168.10.1
SubnetMask	Choices are: • 255.255.255.0 • 255.255.255.128 • 255.255.255.192 • 255.255.255.224 • 255.255.255.240 • 255.255.255.248	Choice	RW	255.255.255.0
DHCP Server Settings (**DeviceInfo.LAN.DHCPServer**)				
Enable	Enable the DHCP Server	Bool	RW	true
ClientAddressRangeStart	Starting value of the 4th (or least significant) byte of the IP address to be assigned by DHCP server. Note that the first three bytes of the IP address is dictated by the value of the Router's LAN side IP Address	UINT	RW	100
MaximumClients	Maximum number of clients to serve	UINT	RW	50
AddressLeaseTime	Address lease time in minutes	UINT	RW	1440
LocalDomainName	The domain name of the LAN	String	RW	

DHCP Reservation Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
DHCP Reservation n (**DeviceInfo.LAN.DHCPServer.ReservationList.**n**.**) for n = 1, 2, …, 20				
Enable	Enable this reservation	Bool	RW	false
Name	A user friendly name to identify this client	String	RW	
MACAddress	MAC Address in "xx:xx:xx:xx:xx:xx" format	String	RW	
IPAddress	The 4th (the least significant) bytes of the IP Address to be reserved. Note that the upper 3 bytes of the IP address is determined by the router's LAN side subnet	String	RW	

Firewall and DMZ Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
DMZ Settings (**DeviceInfo.LAN.DMZ.**)				

OBIHAI
technology, inc.

Enable	Enable DMZ	Bool	RW	`false`
HostIPAddress	The 4th (or least significant) byte of the IP Address of a host server to be put into DMZ of the router. Note that the upper 3-bytes of the IP address is determined by the subnet of the router	UINT	RW	
Firewall Settings (DeviceInfo.Firewall.)				
Enable	Enable the firewall	Bool	RW	`false`
NATRedirection	Allow NAT Redirection (applicable only when firewall is enabled)	Bool	RW	`false`
DRDOSAttackProtection	Enable protection against Distributed Reflection Denial of Service	Bool	RW	`false`
VPNPassThrough	Allow VPN (L2TP, PPTP and IPSEC) traffic to pass through. It is valid when firewall is enabled	Bool	RW	`true`

Port Forwarding Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
Port Forwarding *n* (DeviceInfo.LAN.PortForward.Rule.*n*.) for *n* = 1, 2, …, 20				
Enable	Enable this port forwarding rule	Bool	RW	`false`
Description	Description of the rule	String	RW	
Protocol	The protocols that this rule will be applied to. Choices are: • `TCP` • `UDP` • `Both`	Choice	RW	`TCP`
StartingPortNumber	The first port number in the range of ports ended with the value of EndingPortNumber to forward to the destination	UINT	RW	
EndingPortNumber	The last port number in the range of ports started with StartingPortNumber to forward to the destination	UINT	RW	
ServerIPAddress	The 4th (or least significant) byte of the LAN side host IP address to which the traffic should be forwarded to based on this rule. Note the upper 3 bytes of the IP address are determined by the router's LAN side IP Address	UINT	RW	
DestPortNumber	The port number on the LAN side host to which the traffic should be forwarded based on this rule	UINT	RW	

QoS Settings Web Page

This page shows all the network related settings, as shown below.

Parameter	Description	Type	Access	Default Setting
QoS General Settings (DeviceInfo.Qos.BandwidthControl.)				
Enable	Enable QoS	Bool	RW	`false`
UpStreamBandwidth	Upstream bandwidth limit in kbps when	UINT	RW	`2048`

OBIHAI
technology, inc.

	using the Primary (Ethernet) interface			
RestrictedBandwidth	The bandwidth limit in kbps allocated to the Restricted Class when using the Primary (Ethernet) interface. The Restricted Class has the higher priority than any other priority classes. This value must be smaller than **UpStreamBandwidth**	UINT	RW	512
UpStreamBandwidth2	Upstream bandwidth limit in kbps when failing over to the secondary WAN (LTE) interface	UINT	RW	2048
RestrictedBandwidth2	The bandwidth limit in kbps allocated to the Restricted Class when failing over to the secondary WAN (LTE) interface. The Restricted Class has the higher priority than any other priority classes. This value must be smaller than **UpStreamBandwidth2**	UINT	RW	512
Priority Class Bandwidth Allocation (DeviceInfo.Qos.BandwidthControl.Queue.)				
High	Bandwidth weighting factor when allocating bandwidth according to DSCP values mapped into this priority class. The choices are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	5
Medium	Bandwidth weighting factor when allocating bandwidth according to DSCP values mapped into this priority class. The choices are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	3
Low	Bandwidth weighting factor when allocating bandwidth according to DSCP values mapped into this priority class. The choices are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	2
DSCP to Priority Class Mapping (DeviceInfo.Qos.BandwidthControl.DSCPMapping.)				
N for N = 0, 1, 2, …, 63	DSCP Code N to Priority Class mapping. Choices are: • Restricted • High • Medium • Low	Choice	RW	Medium

System Management Web Pages

Auto Provisioning Web Page

This page shows all the parameters related to remote provisioning of the device, as shown in the following table. Provisioning is an important topic for deployment by service providers. Please refer to this document for details on OBi Device provisioning (http://obihai.com/docs/OBiProvisioningGuide.pdf)

Parameter	Description	Type	Access	Default Setting
Auto Firmware Update (**X_DeviceManagement.FirmwareUpdate.**)				
Method	Current operational method of auto firmware updating. Available choices are: • `Disabled` = Do not check for f/w upgrade from **FirmwareURL** • `System Start` = Check for f/w upgrade from **FirmwareURL** just once on system start • `Periodically` = Check for f/w upgrade from **FirmwareURL** on system start, and then periodically at the interval specified in the Interval parameter • `Time of Day` = Check once at the given **TimeofDay** value Note: First f/w upgrade check on system start will be performed after a random delay of 0-30s	Choice	RW	`Disabled`
TimeofDay	Time of the day in "hh:mm[+rr]" format, valid when method is set to "Time of Day	String	RW	`00:00+30`
Interval	When **Method** is set to `Periodically`, this is the number of seconds between each checking of f/w upgrade check from **FirmwareURL**. If value is `0`, device checks once only on system start (i.e., equivalent to setting **Method** to `System Start`)	UINT	RW	`0`
RandomDelayRange	The range of delay in seconds inserted before the first attempt only. The minimum value shall be 0	UINT	RW	`30`
Suspend	Suspend Firmware update until cancelled	Bool	RW	`false`
FirmwareURL	URL of firmware package. URL must include scheme. Supported schemes are http:// and tftp://	String	RW	

	Choices are: • `A Record Only` • `SRV Record Only` • `Try Both`	Choice	RW	`A Record Only`
DnsLookupType				
DnsSrvPrefix	Choices are: • `No Prefix` • `With Prefix` • `Try Both`	Choice	RW	`No Prefix`
Username	Optional Username for authentication if URL scheme is http://	String	RW	
Password	Optional Password for authentication if URL scheme is http://	String	RW	
ITSP Provisioning (X_DeviceManagement.ITSPProvisioning.)				
Method	Current operational method of Provisioning. Available choices are: • `Disabled` = Do not download from **ConfigURL** • `System Start` = Download from **ConfigURL** just once on system start • `Periodically` = Download from **ConfigURL** on system start, and then periodically at the interval specified in the Interval parameter • `Time of Day` = Once a day at the given **TimeofDay** value Note: First download on system start will be performed after a random delay of 30 – 90s. If there is a firmware update scheduled at the beginning. Or a random delay of 10- 70s..	Choice	RW	`System Start`
Interval	When **Method** is set to `Periodically`, this is the number of seconds between download from **ConfigURL**. If value is `0`, device downloads once only on system start (i.e., equivalent to setting **Method** to `System Start`)	UINT	RW	`0`
TimeofDay	Time of the day in "hh:mm[+rr]" format, valid when method is set to `Time of Day`	String	RW	`00:00+30`

		String	RW	`tftp://$DHCPOPT66/$MAC.xml`
ConfigURL	URL of config file			
DnsLookupType	Choices are: • `A Record Only` • `SRV Record Only` • `Try Both`	Choice	RW	`A Record Only`
DnsSrvPrefix	Choices are: • `No Prefix` • `With Prefix` • `Try Both`	Choice	RW	`No Prefix`
Override	Define what local settings can be overridden by this provisioning. Choices are: • `All` • `All except user settings`	Choice	RW	`All`
GPRM0 to GPRM7	Non-volatile generic parameters which can be referenced in other parameters, such as **ConfigURL**	String	RW	
TPRM0 to TPRM3	Temporary variables used in scripts for **ConfigURL**. Please refer to device provisioning guide for examples on how to use these variables	String	RW	
OBiTALK Provisioning (**X_DeviceManagement.Provisioning.**)				

OBIHAI technology, inc.

Method	Current operational method of Provisioning. Available choices are: • `Disabled` = Do not download from **ConfigURL** • `System Start` = Download from **ConfigURL** just once on system start • `Periodically` = Download from **ConfigURL** on system start, and then periodically at the interval specified in the **Interval** parameter • `Time of Day` = Once a day at the given **TimeofDay** value Note: First download on system start will be performed after a random delay of 30 – 90s. If there is a firmware update scheduled at the beginning. Or a random delay of 10- 70s..	Choice	RW		`System Start`
Interval	When **Method** is set to `Periodically`, this is the number of seconds between download from **ConfigURL**. If value is `0`, device downloads once only on system start (i.e., equivalent to setting **Method** to `System Start`)	UINT	RW	`0`	
TimeofDay	Time of the day in "hh:mm[+rr]" format, valid when method is set to `Time of Day`	String	RW	`00:00+30`	
ConfigURL	URL of config file	String	RW		
DnsLookupType	Choices are: • `A Record Only` • `SRV Record Only` • `Try Both`	Choice	RW		`A Record Only`
DnsSrvPrefix	Choices are: • `No Prefix` • `With Prefix` • `Try Both`	Choice	RW	`No Prefix`	
GPRM0 to GPRM7	Non-volatile generic parameters which can be referenced in other parameters, such as **ConfigURL**	String	RW		
TPRM0 to TPRM3	Temporary variables used in scripts for **ConfigURL**. Please refer to device provisioning guide for examples on how to use these variables	String	RW		
User Defined Macro n (**UDMn$**) (**X_DeviceManagement.X_UserDefinedMacro.n.**) for n = 0, 1, 2, 3					
Value	Any plain text, or reference to another parameter's full TR-104 name prepended by a **$**	String	RW		
ExpandIn	A comma separated list of parameters that are allowed to use this macro expansion. Each parameter must be specified	String	RW		

OBIHAI
technology, inc.

	using its full TR-104 name. Specify ANY to allow it in all parameters.			
SyntaxCheckResult	This is read only status value regarding the syntax of the UDM. "Pass" means that this UDM is valid. Otherwise, it shows the syntax error detected by the device either in the Value or ExpandIn parameters of the UDM.	String	RW	

$MACRO Expansion Supported by the OBi Device

Macro Name	*Description*	*Where It Can Be Used*
MAC	Device MAC address, such as 9CADEF000000	ANY
MACC	Device MAC address with colon, such as 9C:AD:EF:00:00:00	ANY
mac	Device MAC address lower case, with colon, such as 9c:ad:ef:00:00:00	ANY
FWV	Firmware version, such as 1.0.3.1626	ANY
HWV	Hardware version, such as 2.8	ANY
IPA	Device current IP Address, such as 192.168.15.100	ANY
DM	Device Model Name, such as OBi110	ANY
DMN	Device Model Number, such as 110	ANY
OBN	Device OBi Number, such as 200123456	ANY
DSN	Device S/N, such as 88B01NA00000	ANY
GPRMn n = 0–7	Value **Auto Provisioning**::**GPRM**n	**Auto Provisioning**::**ConfigURL**, **Auto Firmware Update**::**FirmwareURL**
TPRMn n = 0-3	Value **Auto Provisioning**::**TPRM**n	**Auto Provisioning**::**ConfigURL**, **Auto Firmware Update**::**FirmwareURL**
UDMn n = 0-3	Value of **User Define Macro** n::**Value**	The value of **User Define Macro** n::**ExpandIn**

Zero-Touch, Massive Scale Remote Provisioning:

OBi ZT or Zero Touch provisioning is a system level approach to deploying and maintaining thousands or millions of OBi devices with high security and control at the device level down to the individual parameter provisioned on each device. Please contact sales@obihai.com for information regarding the capability, process and practice of using OBi ZT Provisioning.

Device Admin Web Page

This page includes the following configuration parameters.

Parameter	Description	Type	Access	Default Setting
Web Server (**X_DeviceManagement.WebServer.**)				
Port	Web Server Port Number	UINT	RW	80
AdminPassword	Administrator Password, case sensitive	String	RW	admin
UserPassword	User Password, case sensitive	String	RW	user
AccessFromWAN	Enable access to the Web Server from the WAN side (when the device operates as a router)	Bool	RW	false
IVR (**X_DeviceManagement.IVR.**)				
Enable	Enable IVR for local configuration	Bool	RW	true
Password	IVR access password (must be all digits)	String	RW	
Syslog (**X_DeviceManagement.Syslog.**)				
Server	IP address or domain name of the Syslog Server where the device sends syslog debug messages to. If the value is blank, syslog is disabled	String	RW	
Port	Syslog Server Port Number	UINT	RW	514
Level	Syslog Message Level. Choices are: 0, 1, 2, 3, 4, 5, 6, 7	Choice	RW	7
TAG	An arbitrary string to add to the beginning of each syslog message		RW	
HTTP Client (**X_DeviceManagement.HTTPClient.**)				
UserAgent	Value of the User-Agent header in all HTTP Requests which are used in firmware upgrade and auto provisioning. Note: Many ITSP relies on this value to tell the make/model of the device in order to return the proper result (such as a configuration file in the context of device provisioning).	String	RW	$DM
TimeOut	Timeout in seconds for an HTTP transaction to complete	UINT	RW	600

Device Update Web Page

There are five different functions offered on this page. These functions are described below.

Firmware Update

You may upgrade the firmware for your OBi device from the device configuration web page. The firmware file with which you want to upgrade the device must be stored locally on a computer from which you can access with a web browser.

Follow these steps to upgrade:

Step 1: Select the, "System Management – Device Update" menu on the side panel of the web page.

Step 2: Specify the path of the firmware file by clicking the, "Select file to upgrade firmware" box or pressing the, "Browse" button in the Firmware Update section of the page. This will present a file browser window where you can navigate to and select the firmware file.

Step 3: Upon selection of the firmware file, press the "Update" button to start the upgrade process.

The entire process will take about 30 seconds to complete. Note that you MUST NOT disconnect the power from the device during this procedure. If the new firmware is upgraded successfully, the OBi device will reboot automatically to start running the new firmware. Otherwise the page will show an error message explaining why upgrade has failed.

Possible Error Messages on Firmware Update Failure:

Error Message	Description	Suggested Solution
Firmware Package Checksum Error	A corrupted Firmware package file has been used for the update.	Check the file and / or re-download the firmware package and try again.
System Is Busy	The OBi device is busy because one of the phone services is in an active call or device provisioning is in progress.	Try to update again later
Firmware Is Not Modified	The OBi device is already running the same firmware as the one selected for update.	No need to upgrade.

Backup (Customized) AA User Prompts

Up to 20 individual prompts may be recorded through the device IVR interface (see *Telephone-IVR-Based Local Configuration* section). These prompts may be backed up into a single file from the web browser. The default name of the file is "backupaa.dat". The backup file also includes the annotations entered for each recorded prompt.

To restore an AA prompt file onto an OBi, do it exactly like a firmware upgrade via the web browser but provide the device with the prompt file instead of a firmware file. The OBi can detect from the file header that you are trying to upload a prompt file and process the file accordingly. *Warning: All the existing prompts in the device will be removed first when applying the backup file; this process cannot be undone*.

Backup Configuration

The current configuration of the OBi device can be backed up and stored as a file in XML format at a user specified location. The default name of the file is "backup*xxxxxxxxxxxx*.xml", where the *xxxxxxxxxxxxx* represents the MAC address of unit.

When backing up a device's configuration, you may select the following three options before selection of the "Backup".

Option	Description	Default Setting
Incl. Running Status	If checked, the value of all status parameters will be included in backup file. Otherwise, status parameters are excluded from the backup	No
Incl. Default Value	If checked, the default value of parameters will be included in the backup file. Otherwise, default values are excluded from the backup	No
Use OBi Version	If not checked, the backup file uses XML tags that are compliant with TR-104 standard. Otherwise, the backup file will be stored in an OBi proprietary format where the XML tags are not compliant with TR-104; but the file size will be smaller and the file will be more readable	No

When the file browser window pops up for, you can change the filename and choose the location to save the backup file. Note that different web browser might handle this differently. If the operation is blocked due to the security setting of the web browser, you should change the security setting temporarily to allow this operation to complete.

Restore Configuration

When restoring the configuration to a previous backup copy, you will need to specify the backup file you want to restore to by selecting the "Browse" button in the Restore Configuration section of the web page. Then, select the "Restore" button to start the process. The OBi device will automatically reboot, after the restoration is complete.

IMPORTANT Note: All passwords and PINs are excluded from the backup file. Hence they will not be available to restore. Call history is excluded from the backup, but can be saved as an XML formatted file separately from the Call History web page.

Reset Configuration (to Factory Default)

The OBi device may be reset to factory default condition. Call history and various statistical information will be removed at the same time. Resetting the device configuration should be used with **extreme caution** as the operation cannot be undone. To do this you press the "Reset" button in the Reset Configuration section. A confirmation window will pop up. The OBi device then proceeds to reset the configuration once you confirm that this is indeed what you want to do. The OBi device will reboot automatically when factory reset is completed.

SIP Service Provider Features of the OBi Device

The following section describes the SIP Service Provider features of the OBi device. Up to **9** SIP accounts (or SIP Trunks) can be configured on the OBi. For the purposed of this document and elsewhere on OBi device web page, documentation and the OBiTALK portal, the term ITSP is used to describe the logical entity providing the SIP Trunk service to the OBi. ITSP stands for Internet Telephony Service Provider. Please note that when the OBi is used in conjunction with an IP PBX, the IP PBX would take the place of the ITSP if it is the entity providing the SIP Trunk account credential and connectivity to the OBi.

Each ITSP configuration is grouped together as an ITSP Profile. There are **9** ITSP profiles available as well and we refer to them as ITSP Profile A, B, C, …, H and I respectively. The SP service account specifics on the other hand are grouped under the heading SPn Service, where n = 1, 2,…, 9. An ITSP Profile includes such parameters as **ProxyServer**, **OutboundProxy**, and **DigitMap**, but does not include account specific parameters. A SP Service includes account specific parameters such as **AuthUserName** (usually the phone number of the account), **AuthPassword**, **CallerIDName**, and **X_ServProfile** (which ITSP Profile to assume). If both SP Service use the same ITSP, then only one ITSP Profile needs to be configured with both SP Services referred to the same profile.

From the OBi device point of view, the SPn Service using ITSP Profile X is enabled with the following minimal settings:

> **ITSP Profile X – SIP**::**ProxyServer** = *a-non-empty-value*
>
> **SPn Service**::**Enabled** = `true`
>
> **SPn Service**::**AuthUsername** = *a-non-empty-value*

where X = A, B, C, …, H or I, and n = 1, 2, …, 8 or 9. Otherwise the service is considered disabled.

SIP Registration

Device can be setup to periodically register with a SIP Proxy Server or SIP Registration Server. SIP Proxy Server and SIP Registration Server can be different, although they are usually the same in practice. SIP Proxy Server is a required parameter that must be configured on the OBi device. The Registration Server is optional and assumed to be the same as the SIP Proxy Server if it is not configured on the device.

The main purpose of registration is to create and maintain a dynamic binding of the SIP account to the device's local contact address. Service provider can also rely on this periodic message to infer if the device is online and functional. Each OBi device takes only one local IP address that is either statically assigned in the device's configuration, or dynamically obtained from a local DHCP server. The SPn services for n = 1 – 9 on the other hand each use a different local contact port for sending and receiving SIP messages (default is 5060, 5061, 5062, …, 5067, and 5068 respectively).

Note that dynamic address binding through periodic registration is not strictly necessary if the local IP address of the device does not change; the device's contact address may be statically configured on the Registration Server.

SIP Outbound Proxy Server

An outbound proxy server can be configured on the device such that all outbound requests are sent via the outbound proxy server instead of directly to the SIP Proxy Server or Registration Server.

DNS Lookup of SIP Servers

When sending out SIP requests to the server, the device looks up the IP address of the server using standard DNS query if the server is specified as a domain name instead of an IP address. If Outbound Proxy Server is configured, it is used instead of the SIP Proxy Server or SIP Registration Server. The resolution of the server domain name into IP address is performed in the following manner:

- Try looking up the name as DNS A Record. If not found.
- Try looking up the name as DNS SRV Record. If not found.
- Try looking up the name as DNS SRV Record with "_sip._udp. " prepended to the hostname. If not found, fail the request.

If the result from the DNS query is a SRV record, the server port is taken from that record also (the server port value configured on the device is ignored). Otherwise, the server port is taken from the configured value or use 5060 if none specified.

NAT Traversal Considerations

If the device sits behind a NAT (typically the case), it can discover the mapped external address corresponding to its local SIP contact address as seen by the server in one of the following ways:

- From the "received=" and "rport=" parameters of the VIA header of the REGISTER response sent by the server; these two parameters tells the device its mapped IP address and port number respectively. This method is used if periodic registration is enabled on the device

- From the response to a STUN binding request the device sent to a STUN server. This method is used by enabling **X_KeepAliveEnable** and setting the **X_KeepAliveMsgType** parameter to "stun". In that case, the STUN server is taken from the **X_KeepAliveServer** parameter, if it is specified. Otherwise, the keep-alive messages are sent to the same server where a REGISTER request would be sent to. The latter is the most effective way of using STUN to discover the mapped external contact address

The device always uses the mapped external contact address in all outbound SIP requests instead of its local contact address if one is discovered by either method discovered above.

SIP Proxy Server Redundancy and Dual REGISTRATION

Server Redundancy specifically refers to the OBi device's capability to a) look for a working server to REGISTER with from among a list of candidates, and b) switch to another server once the server that it currently registers with becomes unresponsive. In other words, DEVICE REGISTRATION MUST BE ENABLED in order to use the server redundancy feature. Other SIP requests, such as INVITE or SUBSCRIBE, are sent to the same server that the device currently registers with.

If Outbound Proxy Server is provided, server redundancy is applied to the Outbound Proxy Server instead of the REGISTRATION server. Server redundancy behavior is enabled by enabling the parameter **ITSP Profile** X **– SIP**::**X_ProxyServerRedundancy** (which is disabled by default).

Another requirement for using the server redundancy feature is that the underlying server must be configured in the device as a domain name instead of an IP address. This allows the OBi to collect a list of candidate servers based on DNS query. The domain name may be looked up as DNS A record or DNS SRV record. For A records, all the IP addresses returned by the DNS server are considered to have the same priority. For SRV records, the hosts returned by the DNS server can be each assigned a different priority.

After a list of candidate servers are obtained, the OBi device will first look for a working server according to the stated priority. A *working server* means one that the device can successfully registers with. This is known as the *Primary Server*. Subsequently, the device maintains registration with the primary server the usual way. However, if no working server is found after traversing the entire list, device takes a short break and repeats the search in the same order.

While maintaining registration with the Primary Server, the OBi will continually attempt to fallback to one of the candidate servers that has higher priority than the primary server, if any. The list of candidate servers that the device is trying to fallback on is known as the *primary fallback list*, which may be empty.

In addition, an OBi device can be configured to maintain a secondary registration with a server that has lower or equal priority than the primary server. Secondary registration can be enabled by setting the parameter **X_SecondaryRegistration** to `true`. If **X_ProxyServerRedundancy** is `false`, however, **X_SecondaryRegistration** does not take any effect. If this feature is enabled, as soon as a primary server is found, the OBi will search for a working secondary server in the same manner from the list of candidate servers that are of lower or equal priority than the primary server. Similarly, once a secondary server is found, the OBi forms a *secondary fallback list* to continually attempt to fallback on if the list is not empty.

The interval for checking the primary fallback list and the secondary fallback list are configured in the parameter **X_CheckPrimaryFallbackInterval** and **X_CheckSecondaryFallbackInterval** respectively. These parameters are specified in seconds and the default value is 60 for both.

Notes:
- Secondary server exists implies primary server exists.
- If the secondary server exists, it immediately becomes the primary server when the current primary server is fails; device then starts searching for a new secondary server if the candidate set is not empty.
- The candidate list may change (lengthened, shortened, priority changed, etc.) on every DNS renewal (based on the entry's TTL). Device will rearrange the primary and secondary servers and fallback lists accordingly, whichever applicable.

If the server redundancy feature is disabled, the device resolves only one IP address from the server's domain name, and will not attempt to try other IP addresses if the server is not responding.

SIP Privacy

The OBi device observes inbound caller privacy and decodes caller's name and number from SIP INVITE requests by checking the FROM, P-Asserted-Identity (PAID for short), and Remote-Party-ID (RPID for short) message headers. All these headers may carry caller's name and number information.

If PAID is present, device takes the name and number from it. Otherwise, it takes name and number from RPID if it is present, or from the FROM header otherwise. RPID, if present, will include privacy setting desired by the caller. This privacy may indicate one of the following options:

- *off* = no privacy requested; the OBi will show name and number.
- *full* = full privacy requested; the OBi will hide both name and number.
- *name* = name privacy requested; the OBi will show the number but hide the name.
- *uri* = uri privacy requested; the OBi will show the name but hide the number.

Regardless, if PAID exists or not, the device always takes the privacy setting from the RPID if it is present in the INVITE request. Note that if the resulting caller name is "Anonymous" (case-insensitive), device treats it as if the caller is requesting full privacy.

For outbound calls, caller's preferred privacy setting can be stated by the device in a RPID header of the outbound INVITE request. To enable this behavior, the parameter **ITSP Profile** X **– SIP**::**X_InsertRemotePartyID** must be set to `true`,

which is the default value of this parameter. OBi only supports two outbound caller privacy setting: privacy=off or privacy=full. The RPID header generated by the device carries the same name and number as the FROM header. If outbound caller-ID is blocked, the device sets privacy=full in RPID, and also sets the display name in the FROM and RPID headers to "Anonymous" for backward compatibility. The device will not insert PAID in outbound INVITE requests. Further, if **ITSP Profile X – SIP**::**X_InsertRemotePartyID** = `true`, the device also includes the SIP header **Privacy: id** in the outbound INVITE request when the user wishes for privacy for the call.

STUN and ICE

The OBi supports standard STUN based on RFC3489 and RFC5389 for passing inbound RTP packets to the device sitting behind NAT's. The parameters that control STUN feature can be found under the section **ITSP Profile X – General**::

- **STUNEnable** – To Enable this feature (default is `false`).
- **STUNServer** – The IP address or domain name of the external STUN server to use. STUN feature will be disabled if this value is blank, which is the default.
- **X_STUNServerPort** – The STUN Server's listening UDP port. Default value `3478` (standard STUN port).

It should be noted that the STUN feature used in this context is only for RTP packets, not SIP signaling packets (which typically does not require STUN). The device's sends out a STUN binding request right before making or answering a call on SPn. If the request is successful, the device decodes the mapped external address and port from the binding response and use them in the m= and c= lines of its SDP offer or answer sent to the peer device. If the request fails, such as STUN server not found or not responding, the call will go on without using external address in the SDP.

Standard RTP requires the use of even number port in the m= line. If the external port is not an even number, device changes the local RTP port and redo STUN, and will continue to do this up to 4 times or until an even external port number is found. If the 4th trial still results in an odd external port number, the call will go on without using external address in the SDP.

OBi supports standard ICE based on RFC5245. ICE is done on a per call basis for automatically discovering which peer address is the best route for sending RTP packets. To enable ICE on the device, set the parameter: **ITSP Profile X – General**::**X_ICEEnable** to `true`. The default is `false`.

Note that ICE would be more effective if STUN is also enabled. However STUN not a requirement for using ICE on the device. If STUN is enabled and an external RTP address different from its local address is discovered, OBi offers two ICE candidates in its SDP:

- The local (host) address (highest priority)
- The external (srflx or server reflexive) address

Otherwise only the local host candidate is shown in the device's SDP. Note that the device uses the srflx address in the m= and c= lines of the SDP if STUN is enabled and successful.

If ICE is enabled and peer's SDP has more than one candidate, device sends STUN requests to each peer candidate from its local RTP port. As soon as it receives a response from the highest priority candidate, device concludes ICE and uses this candidate to communicate with the peer subsequently. Otherwise, the OBi allows up to 5s to wait for the response from all the candidates, and selects the highest priority one that has a response. Once ICE is completed successfully, the device will further apply symmetric RTP concept to determine the peer's RTP address (i.e., send to the address where peer's RTP packets are coming from).

ITSP Driven Distinctive Ringing

OBi device offers 10 ring and 10 call-waiting tone patterns in each ring profile. These patterns are numbered from 1 to 10. Each pattern also comes with a configurable name. A different default ring may be assigned to each trunk on the device.

An ITSP can tell OBi device which ring to use by name for a call routed to SPn by inserting an Alert-Info header in the SIP INVITE sent to the device. The Alert-Info must include a URI. For example:

Alert-Info: http://www.xyz.com/some-folder/bellcore-dr4

When the device receives this, it will look for a ring tone name or call-waiting tone name in the ring profile that matches the Alert-Info URI. Ring tone names are compared case- insensitively. If a match is found, device plays the corresponding ring or call-waiting tone. Otherwise, device plays the default ring.

RTP Statistics – the X-RTP-Stat Header

When ending an established call, OBi device can include a summary of the RTP statistics collected during the call in the SIP BYE request or the 200 response to the SIP BYE request sent by the peer device. The summary is carried in an X-RTP-Stat header in the form of a comma separated list of fields. The reported fields are:

PS=[Number of Packets Sent]
PR=[Number of Packets Received]
OS=[Number of bytes sent]
OR=[Number of bytes received]
PL=[Number of packets lost]
JI=[Jitter in milliseconds]
LA=[Decode latency or jitter buffer size in milliseconds]
DU=[Call duration in seconds]
EN=[Last Encoder Used]
DE=[Last Decoder Used]

For example:

X-RTP-Stat:PS=1234,OS=34560,PR=1236,OR=24720,JI=1,DU=1230,PL=0,EN=G711U, DE=G711U

To enable the X-RTP-Stat feature, the parameter **ITSP Profile** X **– SIP**::**X_InsertRTPStats** must be set to `true`.

Media Loopback Service

The OBi supports the media loopback draft as described in *draft-mmusic-media-loopback-13.txt*. The following media loopback features are supported by the OBi device:

- Loopback modes: loopback-source and loopback-mirror
- Loopback types: rtp-media-loopback and rtp-packet-loopback
- Loopback packet formats:: encaprtp, loopbkprimer

When acts as a loopback mirror, OBi device always sends primer packets so that incoming packets can get through NAT/Firewall. The media loopback feature is controlled by the following parameters (under PHONE Port – Calling Features section):

- **AcceptMediaLoopback** – Enable device to accept incoming call that requests media loopback. Default is `true`.
- **MediaLoopbackAnswerDelay** – The delay in milliseconds before the OBi answers a media loopback call. Default is 0.
- **MediaLoopbackMaxDuration** – The maximum duration to allow for an incoming media loopback call. Default is 0, which means the duration is unlimited.

Note that the device will reject incoming media loopback call if:

- Any PHONE port is off hook.
- Any PHONE port is ringing.
- Any PHONE port is onhook with a call on hold.

Device will terminate an inbound media loopback call already in progress when:

- Any PHONE port is off-hook.
- Any PHONE port is ringing.

To make an outgoing loopback call, user can dial one of the following star codes before dialing the target number:

- *03 – Make a Media Loopback Call.
- *04 – Make a RTP Packet Loopback Call.

Note that outbound Media Loopback Call is not subjected to call duration limit; it will last until the user hangs up or until the called device ends the call.

Service Providers Web Pages

ITSP Profile X – General Web Page (X = A, B, C, D, E, F, G, H, I)

The following configuration parameters are available on this page.

Parameter	Description	Type	Access	Default Setting
ITSP Profile X – General (**VoiceService.1.VoiceProfile.n.**) for X = A, B, …, I corresponding to n = 1, 2, …, 9, respectively				
Name	Human-readable string to identify the profile instance. Maximum Length = 127 characters	String	RW	
SignalingProtocol	Choose among the following list of signalling protocols for this ITSP: • `SIP` • `Google Voice`	Choice	RW	`SIP`
DTMFMethod	Method to pass DTMF digits to peer device. Available choices are: • `Inband` - DTMF tone are sent as inband audio signal • `RFC2833` - DTMF tone events are relayed per RFC2833 • `SIPInfo` - DTMF tones are relayed with SIP INFO request • `Auto` - Method to use based on call setup negotiation (either Inband or RFC2833 may be negotiated) • `SIPInfo+RFC2833` - DTMF tones are relayed by SIP INFO requests and also by RFC2833 RTP Events at the same time	Choice	RW	`Auto`
X_UseFixedDurationRFC2833 DTMF	When relaying DTMF digit events on this trunk using RFC2833, the RFC2833 RTP packets normally will keep streaming for as long as the digit is pressed. With this option set to TRUE, the device sends only one RTP digit event packet with a fixed duration of 150 ms regardless how long the digit has been pressed	Bool	RW	`false`

Parameter	Description	Type	Access	Default Setting
DigitMap	A Digit map to restrict the numbers that be dialed or called with this service. See *OBi Call Routing and Digit Map Section* for a description of digit map syntaxes. Maximum Length = 511 characters	String	RW	(1xxxxxxxxxx|<1>[2-9]xxxxxxxxx|011xx.|xx.)
STUNEnable	Enable device to send a STUN binding request for its RTP port prior to every call	Bool	RW	false
STUNServer	IP address or domain name of the STUN Server to use	String	RW	
X_STUNServerPort	UDP listen port of the STUN Server.	UINT	RW	3478
X_ICEEnable	Enable device to use ICE algorithm to find the best peer RTP address to forward RTP traffic for every call	Bool	RW	false
X_SymmetricRTPEnable	Enable device to apply symmetric RTP behavior on every call: That is, send RTP to peer at the address where incoming RTP packets are received from	Bool	RW	false
Service Provider Info (**VoiceService.1.VoiceProfile.***n***.ServiceProviderInfo.**) for *X* = A, B, …, I corresponding to *n* = 1, 2, …, 9, respectively)				
Name	Human-readable string identifying this service provider. Maximum Length = 127 characters	String	RW	
URL	Website of this service provider. Maximum Length = 127 characters	String	RW	
ContactPhoneNumber	Phone number to contact this service provider. Maximum Length = 31 characters	String	RW	
EmailAddress	Email address to contact this service provider. Maximum Length = 127 characters.	String	RW	

ITSP Profile *X* – SIP Web Page (*X* = A, B, C, D, E, F, G, H, I)

The following configuration parameters are available on this page.

Parameter	Description	Type	Access	Default Setting
ITSP Profile *X* – SIP (**VoiceService.1.VoiceProfile.***n***.SIP.**) for *X* = A, B, …, I corresponding to *n* = 1, 2, …, 9, respectively)				
ProxyServer	Host name or IP address of the SIP proxy server	String	RW	
ProxyServerPort	Destination port to connect to the SIP server	UINT	RW	5060
ProxyServerTransport	Transport protocol to connect to SIP server. The choices are UDP, TCP, or TLS	Choice	RW	UDP
RegistrarServer	Hostname or IP address of the SIP registrar. If a value is specified, device sends REGISTER to the	String	RW	

	given server; otherwise REGISTER is sent to **ProxyServer**			
RegistrarServerPort	Destination port to connect to SIP registrar	UINT	RW	`5060`
UserAgentDomain	CPE domain string. If empty, device uses **ProxyServer** as its own domain to form its AOR (Address Of Record) or Public Address when constructing SIP messages (for example, in the FROM header of outbound SIP Requests). Note: If **SP***n* **Service::URI** is specified, additional rules applied in forming the AOR. See description of **URI** parameter for more details and examples	String	RW	
OutboundProxy	Host name or IP address of the outbound proxy. Outbound proxying is disabled if this parameter is blank.	String	RW	
OutboundProxyPort	Destination port to be used in connecting to the outbound proxy	UINT	RW	`5060`
X_OutboundProxyTransport	A different SIP transport may be used by the OutboundProxy. The available choices are: • `UDP`: • `TCP`: • `TLS`: • `Follow ProxyServerTransport`:	Choice	RW	`Follow ProxyServerTransport`
X_UserAgentContactFollowP roxyServerTransport	If enabled, the user agent should use a Contact and Via transport that agrees with ProxyServerTransport	Bool	RW	`false`
X_BypassOutboundProxyInC all	Bypass the outbound proxy after initial call setup	Bool	RW	`false`
RegistrationPeriod	Nominal interval between device register in seconds	UINT	RW	`60`
X_RegistrationMargin	Specifies the margin to renew SIP registration before it expires. The default is to renew at half-time before the next expiration if the expires value is less than 1200s; or 600s before expiration otherwise. You can specify here the number of seconds before expiration to renew registration explicitly (such as 10 or 20), or as a fraction of the current register expires value (such as 0.1 or 0.25)	String	RW	
TimerT1	Value of SIP timer T1 in ms	UINT	RW	`500`
TimerT2	Value of SIP timer T2 in ms	UINT	RW	`4000`
TimerT4	Value of SIP timer T4 in ms	UINT	RW	`5000`
TimerA	Value of SIP timer A in ms	UINT	RW	`500`
TimerB	Value of SIP timer B in ms	UINT	RW	`32000`

TimerD	Value of SIP timer D in ms	UINT	RW	32000	
TimerE	Value of SIP timer E in ms	UINT	RW	500	
TimerF	Value of SIP timer F in ms	UINT	RW	32000	
TimerG	Value of SIP timer G in ms	UINT	RW	500	
TimerH	Value of SIP timer H in ms	UINT	RW	32000	
TimerI	Value of SIP timer I in ms	UINT	RW	5000	
TimerJ	Value of SIP timer J in ms	UINT	RW	32000	
TimerK	Value of SIP timer K in ms	UINT	RW	5000	
InviteExpires	Invite request Expires header value in seconds	UINT	RW	60	
ReInviteExpires	Re-invite Expires header value in seconds	UINT	RW	10	
RegisterExpires	Register Expires header value in seconds (not used at the moment)	UINT	RW	3600	
RegistersMinExpires	Register Min-Expires header value in seconds (not used at the moment)	UINT	RW	15	
RegisterRetryInterval	Register retry interval in seconds	UINT	RW	30	
X_RegisterRetryResponseCodes	A set of rules to control how many seconds to wait before retrying register after a specific failure response. The rules are specified with a digitmap string; the value after the letter w designates the number of seconds to wait. A range may be specified with two numbers with a – in between such that a random delay within that range is used.	String	RW	(<40[17]:w120>|<40[34]:w120>|<99[01]:w120>|<120-200>|[4-9]xx)	
DSCPMark	Diffserv code to use for outgoing SIP packets. Valid range: 0-63	UINT	RW	0	
X_SpoofCallerID	Allow outbound Caller ID spoofing. If set to Yes, device will attempt to set the caller-id name and userid field in the FROM header to that of a remote caller in the case of a bridged call (from another trunk, such as PSTN Line or another SP Service). Otherwise, device always its own account information to form the FROM header.	Bool	RW	false	

	Note that most service provider will not allow originating a call if the FROM header field does not match the account credentials. Enable this option only if you are sure that the service provider allows it. For example, an IP PBX may allow it.			
X_UseRefer	Enable the use of SIP REFER for call transfer. If disabled, device will bridge the call instead when performing a call transfer (which consume some resources on the device)	Bool	RW	false
X_ReferAOR	Enable the use the target's AOR (Address of Record or public address) in Refer-To header of SIP REFER. If disabled, the target's Contact will be used instead	Bool	RW	true
X_HoldReferee	Hold the Referee before a blind transfer if the call is not placed on hold. This may allow reconnecting with the Referee if the blind call transfer fails	Bool	RW	false
X_Use302ToCallForward	Enable the use of 302 response to INVITE for call forward. If disabled, device will bridge the call legs instead when forwarding a call (and will consume some resources on the device)	Bool	RW	true
X_UserAgentName	If a value is specified, device includes a User-Agent header in all SIP Requests, or a Server header in all SIP responses, that contains exactly the given value	Bool	RW	OBIHAI/${DM}-${FWV}
X_ProcessDateHeader	Enable the device to decode the DATE header sent by the ITSP in a 200 response to its REGISTER. The DATE header specifies the current GMT time and the device can use to adjust its local time and date without relying on NTP	Bool	RW	true
X_InsertRemotePartyID	Enable the device to include a Remote-Party-ID header in its outbound SIP INVITE to indicate to the ITSP the caller's preferred privacy setting (either full or none)	Bool	RW	true
X_InsertPrivacyHdr	Insert a 'Privacy:id' header in outbound INVITE for anonymous calls	Bool	RW	false
X_UseAnonymousFROM	Enable the use of "sip:anonymous@localhost" in FROM header of SIP INVITE when attempting to make an anonymous call	Bool	RW	false
X_SessionRefresh	Enable session refresh signalling (with SIP Re-INVITE) during a connected call. This allows the OBi to detect if the connection with the peer is broken abnormally so it can release the call. Disable this option if the ITSP does not support Re-INVITE sent from the client device.	Bool	RW	true
X_SessionTimer	Enable standard session timer behaviour based on RFC4028	Bool	RW	false
X_SessionExpires	Session Expires before value. If session refresh is enabled, OBi will refresh half-time before the session expires.	UINT	RW	20

X_AccessList	A comma separated list of IP addresses such that the device only accepts SIP requests coming from one of the given addresses. If the list is empty, the device accepts SIP requests from any IP address	String	RW	
X_InsertRTPStats	Enable the device to include a X-RTP-Stat header in a BYE request or 200 response to BYE request at the end of an established call. This header contains a summary of RTP statistics collected during the call.	Bool	RW	true
X_MWISubscribe	Enable this option to have the device SUBSCRIBE to the message-summary event package to support MWI and VMWI service. Note that device handles NOTIFY of this event package regardless **MWISubscribe** is enabled or not	Bool	RW	false
X_MWISubscribeURI	Blank implies to use the same URL as REGISTER for the TO and FROM header as well as the Request-URI Otherwise, if the URI does not contain '@', it is user as the userid field in TO/FROM header as well as the Request-URI, which are otherwise same as REGISTER If the URI contains '@', it is used in the TO and FROM header as well as the Request-URI as is Note that OBi device forms the Request-URI of SUBSCRIBE the same way as the TO header, with an additional port number	String	RW	
X_MWISubscribeExpires	Periodic interval in seconds to renew SUBSCRIBE	UINT	RW	3600
X_RegSubscribe	Enable sending SUBSCRIBE for **reg** event package	Bool	RW	false
X_RegSubscribeExpires	Reg Event Subscription Renewal interval in seconds. Set the value to 0 to disable subscription renewal	UINT	BW	3761
X_NoNonceAuth	Enable sending Authorization header without being challenged first	Bool	RW	false
X_ProxyServerRedundancy	Enable proxy redundancy feature on the device. To use this feature, device registration must be enabled and the SIP Registration Server or Outbound Proxy Server must be configured as a domain name	Bool	RW	false
X_SecondaryRegistration	Enable device to register with a secondary server in addition to the primary server. **X_ProxyServerRedundancy** must be enabled for this parameter to take effect	Bool	RW	false
X_CheckPrimaryFallbackInterval	Interval in seconds at which the device should	UINT	RW	60

	check the primary fallback list of candidate servers			
X_CheckSecondaryFallbackInterval	Interval in seconds at which the device should check the secondary fallback list of candidate servers	UINT	RW	60
X_ProxyFailoverResponseCodes	A list of failure response codes specified in the form of a digitmap string to trigger proxy failover. If only one digitmap is specified, it applies to REGISTER and INVITE requests. If two digitmaps are provided (separated by a comma), the first one applies to REGISTER and the second to INVITE.	String	RW	([5-9]xx)
X_ProxyRequire	If this parameter is not blank, OBi will include a Proxy-Require header stating the value of this parameter in all SIP requests sent to the ITSP	String	RW	
X_MaxForward	Value for the Max-Forward header in all SIP requests sent by the OBi	UINT	RW	70
X_AcceptLanguage	If this parameter is not blank, OBi will include an Accept-Language header stating the value of this parameter in all SIP requests sent to the ITSP.	String	RW	
X_DnsSrvAutoPrefix	Enable this option to let OBi automatically prepend a standard prefix to the domain name when querying DNS Server to resolve the ProxyServer or OutboundProxy name as a SRV record. The standard prefix is _sip._udp. for SIP over UDP, _sip._tcp. For SIP over TCP, and _sip._tls. for SIP over TLS.	Bool	BW	false
X_Support100rel	Enable this option to turn on the support for RFC3262 (reliable provisional SIP responses). If enabled, OBi will announce this support in a SIP Supported header, and will require a caller to use this option if the caller also supports this feature.	Bool	RW	false
X_UserEqPhone	Enable the insertion of user=phone parameter in INVITE Request-URI	Bool	RW	false
X_UseTelURI	Enable the use of tel: in outbound SIP Request-URI and TO-URL	Bool	RW	false
X_CallWaitingIndication	Choices are: • No • Alert-Info	Choice	RW	No
X_DiscoverPublicAddress	Enable this option to let the OBi use the public IP address and port it has discovered as its SIP Contact address	Bool	RW	true
X_UsePublicAddressInVia	Use the public IP address and port discovered by the OBi as the address in VIA header	Bool	RW	false
X_PublicIPAddress	A static public IPv4 address, if specified, will be used by the OBi to form its SIP Contact address	String	RW	
X_UseRport	Enable this option to let the OBi insert a blank rport parameter in the VIA header our outbound SIP messages. This option should be turned off if you are using port forwarding on the external router to route inbound SIP messages to the OBi	Bool	RW	true
X_DetectALG	Enable detection of upstream SIP ALG	Bool	RW	false
X_UseCompactHeader	Enable the use of compact format SIP headers	Bool	RW	false
X_OmitContentLength	Omit Content-Length header if ProxyServerTransport and	Bool	RW	false

OBIHAI
technology, inc.

	X_OutboundProxyTransport are both UDP			
X_FaxPassThroughSignal	Method to signal FAX passthrough to the peer. Available options are: • `ReINVITE` • `RFC2833` • `Auto` • `None` If `None` is selected, FAX passthrough will not be signalled. If Auto is selected, RFC2833 is used if the peer has indicated support in SDP.	Choice	RW	`ReINVITE`
X_IncludeMessageHash	Include a MD5 hash of all the SIP headers in an **X-MD5-Hash** header. A hash of the SDP is also included in an x-md5-hash SDP attribute	Bool	RW	`false`
X_EchoServer	A server that can echo back SIP messages to the device	String	RW	
X_EchoServerPort	The echo server listening port	UINT	RW	`5060`
X_EnableRFC2543CallHold	Enable the device to recognize call hold signalling as used in RFC2543	Bool	RW	`false`

ITSP Profile *X* – RTP Web Page (*X* = A, B, C, D, E, F, G, H, I)

Parameter	Description	Type	Access	Default Setting
ITSP Profile *X* – RTP (**VoiceService.1.VoiceProfile.*n*.RTP.**) for *X* = A, B, …, I corresponding to *n* = 1, 2, …, 9, respectively				
LocalPortMin	Base of port range for tx/rx RTP with this SP	UINT	RW	`16600` (*X*=A) `16800` (*X*=B) `17000` (*X*=C) `17200` (*X*=D) `17400` (*X*=E) `17600` (*X*=F) `17800` (*X*=G) `18000` (*X*=H) `18200` (*X*=I)
LocalPortMax	Top of port range for tx/rx RTP with this SP	UINT	RW	`16798` (*X*=A) `16998` (*X*=B) `17198` (*X*=C) `17398` (*X*=D) `17598` (*X*=E) `17798` (*X*=F) `17998` (*X*=G) `18198` (*X*=H) `18398` (*X*=I)
KeepAliveInterval	Interval in seconds between sending keep alive packet on an RTP channel that is currently in idle (due to call hold for instance). RTP keepalive is disabled if the value of this parameter is set to 0.	UINT	RW	`0`
DSCPMark	Diffserv code for outgoing RTP packets with this SP. Valid range is 0 – 63	UINT	RW	`0`
X_UseSSL	Enable this option to force OBi to send RTP over a SSL channel when the ITSP is Google Voice	UINT	RW	`false`
ITSP Profile *X* – RTCP (**VoiceService.1.VoiceProfile.*n*.RTP.RTCP.**) for *X* = A, B, …, I corresponding to *n* = 1, 2, …, 9, respectively				

Enable	Enable RTCP operation	Bool	RW	false
TxRepeatInterval	Interval in milliseconds between RTCP transmissions	UINT	RW	10000
LocalCName	The local CNAME to use in RTCP message. By default OBi uses account-userid@local-ip-address as the local CNAME.	String	RW	
X_RTCPMux	Enable RTCP-Mux operation (send and receive RTCP on the same local port as the corresponding RTP)	Bool	RW	false
X_VqPublishEnable	Enable VQ report sent to the proxy server using Publish method	Bool	RW	false
X_VqPublishUrl	A Username or URL to send Voice Quality Report using Publish method	String	RW	
X_VqPublishInterval	Interval in seconds between VQ reports; 0 or *an-empty-value* disables periodic reports	UINT	RW	0
X_VqPublishOnSSRCChange	Enable VQ report when SSRC changes	Bool	RW	true

Google Voice™ Service

OBi device includes a native implementation of the Google Talk™ (XMPP) protocol that allows the user to use the Google Voice™ communications service as the SPn service, n = 1 – 9. To enable Google Voice for an SP service, set the underlying **ITSP Profile** X **General**::**Protocol** parameter to `Google Voice` (the default value of this parameter is "SIP"), where X = A, B, …, H, or I. All SP services can be enabled for Google Voice, with a different account on each service.

Google Voice offers a call screening feature such that you must press digit 1 before answering an incoming GV call. OBi device can be setup to automatically do that for you when you pick up the phone. To enable this feature on the device, set the **X_SkipCallScreening** parameter to `true` (default is `false`) (on the device web page, under the SPn Service – CallingFeatures section).

Please note that the codec is limited to G711u only for all calls.

When Google Voice is selected as the protocol, all the other ITSP Profile parameters are ignored except the DigitMap parameter. The following SPn Service parameters are also ignored:

- **X_Codec_Profile**, **X_RegisterEnable**, **X_UserAgentPort**, **X_UserAgentPorts**
- **X_SipDebugOption**, **X_SipDebugExclusion**
- **X_AcceptSipFromRegistrarOnly**, **X_NoRegNoCall**
- **X_KeepAliveEnable**, **X_KeepAliveExpires**, **X_KeepAliveServer**, **X_KeepAliveServerPort**, **X_KeepAliveMsgType**, **X_CustomKeepAliveMsg**
- **X_SatelliteMode**, **X_AcceptResync**
- **AuthPassword**, **URI**, **X_ContactUserID**, **X_EnforceRequestUserID**, **CallerIDName**
- **MaxSessions**, **X_AcceptDialogSubscription**, **X_AcceptLinePortStatusSubscription**
- **X_MWIRoute**, **X_VMWIRoute**, **X_SRTP**

The following features are also supported:

- MWI (Message Waiting Indication) and VMWI (Visual Message Waiting Indication) for Google Voicemail.

- Non-Gmail domain in account name for Google Voice Communications Service.

- Accept DTMF input from a Google Talk client entered by the user as text messages (only 0 – 9, *, and # will be recognized by the device).

- Accept the setting of the parameter **ITSP Profile** X **– General**::**DTMFMethod**. The value can be either `InBand` or `RFC2833`. Other values will be reverted to `RFC2833`. Default is `RFC2833`.

- Voice Service Features of the OBi Device.

Voice Services Web Pages

SP*n* Service Web Page (*n* = 1, 2, 3, 4, 5, 6, 7, 8, 9)

The following configuration parameters are available on this page.

Parameter	Description	Type	Access	Default Setting
SPn Service (**VoiceService.1.VoiceProfile.1.Line.***n***.**) for *n* = 1, 2, …., 9				
Enable	Enable this line	Bool	RW	true
X_ServProvProfile	Select a Service Provider Profile for this service. Choices: A, B, C, D, E, F, G, H, I	Choice	RW	A
X_RingProfile	Select a Ring Profile to ring the PHONE port with for incoming calls on this service that are routed to the PHONE port. The ringing pattern will be taken from the given profile. Choices are A, B	Choice	RW	A
X_CodecProfile	Select a Codec Profile for all calls on this service. Choices are A, or B	Choice	RW	A
X_InboundCallRoute	Routing rule for directing incoming calls on this service. The default rule is to send all incoming calls to the PHONE port (ph). See *OBi Call Routing and Digit Map Section* for a description of the syntaxes for specifying this parameter	String	RW	ph
X_AcceptSipFromRegistrar Only	Enable the device behaviour such that it only accepts SIP packets coming from the current registrar IP address	Bool	RW	false
X_RegisterEnable	Enable registration for this line. If set to true, device sends periodic SIP REGISTER to the service provider according to the settings in the ITSP Profile. Otherwise, device does not send any SIP REGISTER for the service	Bool	RW	true
X_NoRegNoCall	Disallow all incoming and outgoing calls if not successfully registered	Bool	RW	false
X_KeepAliveEnable	Enable sending keep alive message. If set to YES, device sends periodic keep-alive messages to the destination specified in X_KeepAliveServer and X_KeepAliveServerPort, at the interval specified in X_KeepAliveExpires. The content of this message is the ascii string "keep-alive\r\n"	Bool	RW	false
X_KeepAliveExpires	Keep alive period in seconds	UINT	RW	15
X_KeepAliveServer	Hostname or IP address of keep alive server	String	RW	
X_KeepAliveServerPort	UDP port of the keep alive server	UINT	RW	5060

X_KeepAliveMsgType	The type of keep alive messages to send out periodically if keep-alive is enabled. It can be one of the following choices: • `keep-alive`: The string "keep-alive" • `empty`: A blank line • `stun`: A standard STUN binding request; device will use the binding response to form its contact address for REGISTRATION • `custom`: use the value of X_CustomeKeepAliveMsg (note: option not available on OBi100/OBi110) • `options`: A SIP OPTIONS message • `notify`: A SIP NOTIFY message	Choice	RW	`keep-alive`
X_CustomKeepAliveMsg	Defines the custom message to be used when X_KeepAliveMsgType is "custom". The value should have the following format: `mtd=NOTIFY;event=<whatever>;user=<anyone>` Where - NOTIFY may be replaced by any other SIP method, such as PING, - event parameter is optional and is only applicable if method is NOTIFY. If event is not specified, the 'keep-alive' event will be used with NOTIFY - user parameter is optional; if not specieifed, the request-uri will not have a userid, and the TO header field will use the same userid as the FROM header (which is the local account userid). If user is specified, it will be used as the userid in the Request-URI and TO header. SIP messages for keep-alive are sent only once without retransmission; response to the SIP messages are ignored by the OBi.	String	RW	
X_UserAgentPort	UDP port where the device sends and listens for SIP messages	UINT	RW	`5060` (n=1) `5061` (n=2) `5062` (n=3) `5063` (n=4) `5064` (n=5) `5065` (n=6) `5066` (n=7) `5067` (n=8) `5068` (n=9)
X_UserAgentPorts	A comma separated list of (up to 10) alternative UDP Ports for tx/rx SIP packets	String	RW	
DirectoryNumber	Directory number associated with this service	String	RW	
X_DefaultRing	Default ring pattern number to ring the PHONE port for	Choice	RW	`1`

	incoming calls on this trunk that are routed to the PHONE port according to the InboundCallRoute of this service. The ring pattern is taken from the selected Ring Profile. Valid choices are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10			
X_CallOnHoldRing	Pattern to ring PHONE port when holding a call on this trunk that has been connected to the PHONE port. Typically this is a very short distinctive ring pattern that serves as a reminder to the user that a call is being on hold. The ring pattern is taken from the selected Ring Profile. Valid choices are: No Ring, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	8
X_RepeatDialRing	The ring pattern number to use to ring the PHONE port when a repeat dial operation on this trunk is successful as the called party is either ringing or answered. Choices are: Valid choices are 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	5
X_BargeInRing	Call Waiting Ring pattern to ring the PHONE port when the incoming call is requesting to barge-in. This is applicable in a call-waiting scenario on the PHONE port. Choices are No Ring, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	4
X_CallParkedRing	Ring pattern to ring the PHONE port only as a reminder that there are some calls parked in the parking lot. This feature is applicable only in an OBiPLUS solution. Choices are No Ring, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	10
X_SIPDebugOption	Enable sending of SIP signaling debug information to the syslog server (if one is configured on the device). Available choices are: • Disable (do not send SIP signaling debug information) • Log All Messages • Log All Except REGISTER Messages	Choice	RW	Disable
X_SipDebugExclusion	A list of SIP methods to exclude from the syslog for this SP service. For example: notify, subscribe	String	RW	
X_SatelliteMode	Enable satellite mode on this trunk. In this mode, the user must explicitly sign on (using * code) to receive phone calls on this trunk. The SIP REGISTER sent by the OBi to the ITSP on this trunk will indicate if the user wants to sign on (and therefore takes over the incoming calls for this account). This feature is only applicable if the service is provided by an OBiPLUS system	Bool	RW	false
X_AcceptResync	Control whether to accept a SIP NOTIFY request with event=resync to trigger a reboot of the device (so it can download new f/w or configuration upon boot up). Available choices are: • no (do not accept resync trigger) • yes with authentication (accept after challenging the sender) • yes without authentication (accept w/o challenging the sender)	Choice	RW	yes without authentication
SPn Service – SIP Credentials (VoiceService.1.VoiceProfile.1.Line.n.SIP.) for n = 1, 2, …., 9				

AuthUserName	The User ID to authenticate to a SIP UAS (User Agent Server) when an outbound SIP request sent by the device is challenged by the UAS with a 401 or 407 Response	String	RW	
AuthPassword	The Password (corresponding to AuthUserName) to authenticate to a SIP UAS (User Agent Server) when an outbound SIP request sent by the device is challenged by the UAS with a 401 or 407 Response	String	RW	

URI	This parameter affects the way the AOR is formed by the device in outbound SIP Requests. The AOR has the format: user@domain.
	If the value of URI is empty, device gets the user portion of its AOR from the AuthUserName, and the domain portion the value of ITSP Profile's UserAgentDomain if it is not empty, or that of the ProxyServer otherwise.
	If the value URI is not empty and does not contain "@", it is used as the user portion of the AOR while the domain portion is formed the usual way.
	If the value of URI contains "@', it is interpreted as a full AOR and device takes it as the AOR as is.
	
	Some Examples:
	1) Let ProxyServer = *sip.myitsp.com*, AuthUserName = *4089991123*, URI=[empty], UserAgentDomain=[empty], then
	  AOR = _4089991123@sip.myitsp.com_
	
	2) Change UserAgentDomain to *users.myitsp.com*, then
	  AOR = _4089991123@users.myitsp.com_
	
	3) Change URI to bobdylan, then
	  AOR = _bobdylan@users.myitsp.com_
	
	4) Change URI to _bobdylan@superusers.myitsp.com_, then
	  AOR = _bobdylan@superusers.myitsp.com_
	
	Note: In all cases, device uses **AuthUserName** and **AuthUserPassword** to compute authorization if challenged by a 401 or 407 response.
		String	RW	

X_ContactUserID	An alternative userid to be used in Contact header. Enter `Random` to let the device generate a random one.	String	RW	
X_EnforceRequestUserID	Enforce incoming INVITE request userid to match **AuthUserName** or **ContactUserID**	Bool	RW	`false`
SP*n* Service – SIP Credentials (VoiceService.1.VoiceProfile.1.Line.*n*.CallingFeatures.) for *n* = 1, 2, …., 9				
CallerIDName	Display name to identify the subscriber. The display name field is usually inserted in a FROM header in outbound SIP requests (such as INVITE) for the purpose of displaying a Caller ID Name on the recipient's device.	String	RW	
MaxSessions	The maximum number of simultaneous calls that may be established on this service	UINT	RW	`2`
CallForwardUnconditional Enable	Enable call forwarding of all calls unconditionally by the	Bool	RW	`false`

	device. If **CallForwardUnconditionalNumber** is blank, this parameter is treated as if it has been set to `false`. Note: It is possible for a user to set this parameter from the phone using a Star Code			
CallForwardUnconditional Number	Directory number to forward all incoming calls on this service unconditionally. Maximum Length is 127 characters. Note: It is possible for a user to set this parameter from the phone using a Star Code	String	RW	
CallForwardOnBusyEnable	Enable call forwarding of all incoming calls when the device is busy. If **CallForwardOnBusyNumber** is blank, this parameter is treated as if it has been set to *No*. Device is considered busy if one of the following conditions holds: This service already reaches the limit of simultaneous calls as specified in MaxSessions DND (Do Not Disturb) Service is enabled on this service If the call is routed to the PHONE port where the phone is in a busy state (such as ringing, dialing, playing reorder, or already having 2 calls in progress) Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	`false`
CallForwardOnBusyNumbe r	Directory number to forward all incoming calls on this service when the device is busy. Maximum Length is 127 characters. Note: It is possible for a user to set this parameter from the phone using a Star Code	String	RW	
CallForwardOnNoAnswerE nable	Enable call forwarding of all incoming calls when the call is not answered after a period as specified in **CallForwardOnNoAnswerRingCount**. If **CallForwardOnNoAnswerNumber** is blank, this parameter is treated as if it has been set to No. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	`false`
CallForwardOnNoAnswerN umber	Directory number to forward all incoming calls when the call is not answered after a period specified in **CallForwardNoAnswerRingCount** Note: It is possible for a user to set this parameter from the phone using a Star Code	String	RW	
CallForwardOnNoAnswerRi ngCount	Number of rings to be considered by the device as no answer to an incoming call. Note: 1 ring is approximately 6s	UINT	RW	`2`
BlockedCallers	A comma separated list of up to 10 caller numbers to block from calling this service	String	RW	
MWIEnableMask	Enable Message Waiting Indication from this service on one or more phone ports. It is specified as a bit mask,	UINT	RW	`255`

Parameter	Description	Type	Access	Default Setting
	such that bit 0 for Phone 1, bit 1 for phone 2, etc.			
X_VMWIEnableMask	Enable Visual Message Waiting Indication for this service on one or more phone ports. It is specified as a bit mask, such that bit 0 for Phone 1, bit 1 for phone 2, etc.	UINT	RW	255
X_MWIRoute	SIP/NOTIFY Routing Rules to enable MWI signals on MWI Notifications	String	RW	
X_VMWIRoute	SIP/NOTIFY Routing Rules to enable VMWI signals on MWI Notifications	String	RW	
MessageWaiting	This is a **state** rather than a configuration parameter, that indicates if there are any new messages for this subscriber on the service provider's voicemail system	Bool	RW	false
AnonymousCallBlockEnable	Enable blocking of Anonymous Calls on this service. Anonymous calls are rejected with a SIP 486 (Busy) response and Call Forward On Busy service is not applied. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
AnonymousCallEnable	Enable masking of Caller-ID information for all outgoing calls. If enabled, the called party should perceive the call as coming from an anonymous caller. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
DoNotDisturbEnable	Enable Do Not Disturb Service. If enabled, all incoming calls on this service are treated as if the device is busy. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
X_BridgedOutboundCallMaxDuration	Limit on the call duration in seconds for all outbound calls that are bridged from the same or another trunk. A *empty-value* or 0 value implies the call duration is not limited.	UINT	RW	
X_AcceptDialogSubscription	Enable the device to accept SUBSCRIBE to this trunk's dialog event package	Bool	RW	false
X_SkipCallScreening	Enable the device to automatically skip call screening when the underlying ITSP is Google Voice	Bool	RW	true
X_SRTP	This is a drop down list with 3 choices: • Disable SRTP = Do not use SRTP for all calls; the call will fail if the peer insists on using SRTP only • Use SRTP Only = Require all calls to use SRTP; the call will fail if the peer does not support SRTP • Use SRTP When Possible = Use SRTP for a call if the peer supports SRTP; otherwise fallback to use regular unencrypted SRTP	Choice	RW	Disable SRTP

OBiTALK Service Web Page

Parameter	Description	Type	Access	Default Setting
OBiTALK Service Settings (**VoiceService.1.X_P2P.1.**)				

Enable	Enable the OBiTALK Service (the built-in free voice service that comes with every OBi Device)	Bool	RW	`true`
LocalPort	The UDP or TCP port used by device to send and listens for OBiTALK messages	UINT	RW	`10000`
TryMultiplePorts	Enable the unit to try a few random UDP ports until it can successfully join the OBiTALK network	Bool	RW	`true`
DisplayName	Display name to identify the subscriber, for the purpose of displaying a Caller ID Name on the recipient's device	String	RW	
DigitMap	Digit map to restrict numbers that can be dialed or called with this service. See *OBi Call Routing and Digit Map* section for a description of the syntaxes for specifying a Digit Map.		RW	`(<ob>xxxxxxxxx|obxxxxxxxxx)`
InboundCallRoute	Routing rule for directing incoming calls on this service. The default rule is to send all incoming calls to the PHONE port (ph). See *OBi Call Routing and Digit Map* section for a description of the syntaxes for specifying this parameter	String	RW	`ph,ph2`
RingProfile	Select a Ring Profile to ring the PHONE port with when an incoming call is routed to the PHONE port. Choices are `A`, `B`	Choice	RW	`A`
CodecProfile	Select a Codec Profile to be used for all calls on this service. Choices are `A`, `B`	Choice	RW	`A`
DefaultRing	Default ring pattern number to ring the PHONE port for incoming calls on this trunk that are routed to the PHONE port according to the InboundCallRoute of this service. The ring pattern is taken from the selected Ring Profile. Valid choices are `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`	Choice	RW	`2`
CallOnHoldRing	Pattern to ring PHONE port when holding a call on this trunk that has been connected to the PHONE port. Typically this is a very short distinctive ring pattern that serves as a reminder to the user that a call is being on hold. The ring pattern is taken from the selected Ring Profile. Valid choices are: `No Ring`, `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`	Choice	RW	`8`
RepeatDialRing	The ring pattern number to use to ring the PHONE port when a repeat dial operation on this trunk is successful as the called party is either ringing or answered. Valid choices are: `1`, `2`, `3`, `4`, `5`, `6`, `7`, `8`, `9`, `10`	Choice	RW	`4`
DTMFMethod	Method to pass DTMF digits to peer device. Available choices are: • `Inband` - DTMF tone are sent as inband audio signal • `RFC2833` - DTMF tone events are relayed per RFC2833 • `SIPInfo` - DTMF tones are relayed with SIP INFO	Choice	RW	`Auto`

	request • `Auto` - Method to use based on call setup negotiation (either Inband or RFC2833 may be negotiated)			
UseFixedDurationRFC2833 DTMF	When relaying DTMF digit events on this trunk using RFC2833, the RFC2833 RTP packets normally will keep streaming for as long as the digit is pressed. With this option set to TRUE, the device sends only one RTP digit event packet with a fixed duration of 150 ms regardless how long the digit has been pressed	Bool	RW	`false`
OBiTALK Service – Calling Features (**VoiceService.1.X_P2P.1.CallingFeatures.**)				
CallForwardUnconditionalE nable	Enable call forwarding of all calls unconditionally by the device. If **CallForwardUnconditionalNumber** is blank, this parameter is treated as if it has been set to *No*. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	`false`
CallForwardUnconditional Number	Directory number to forward all incoming calls on this service unconditionally. Maximum Length is 127 characters. Note: It is possible for a user to set this parameter from the phone using a Star Code	String	RW	
CallForwardOnBusyEnable	Enable call forwarding of all incoming calls when the device is busy. If **CallForwardOnBusyNumber** is blank, this parameter is treated as if it has been set to *No*. Device is considered busy if one of the following conditions holds: This service already reaches the limit of simultaneous calls as specified in **MaxSessions** DND (Do Not Disturb) Service is enabled on this service If the call is routed to the PHONE port where the phone is in a busy state (such as ringing, dialing, playing reorder, or already having 2 calls in progress) Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	`false`
CallForwardOnBusyNumbe r	Directory number to forward all incoming calls on this service when the device is busy. Maximum Length is 127 characters. Note: It is possible for a user to set this parameter from the phone using a Star Code	String	RW	
CallForwardOnNoAnswerE nable	Enable call forwarding of all incoming calls when the call is not answered after a period as specified in **CallForwardOnNoAnswerRingCount.** If **CallForwardOnNoAnswerNumber** is blank, this parameter is treated as if it has been set to No. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	`false`
CallForwardOnNoAnswerN umber	Directory number to forward all incoming calls when the call is not answered after a period specified in **CallForwardNoAnswerRingCount**	String	RW	

	Note: It is possible for a user to set this parameter from the phone using a Star Code			
CallForwardOnNoAnswerRingCount	Number of rings to be considered by the device as no answer to an incoming call. Note: 1 ring is approximately 6s	UINT	RW	2
BlockedCallers	A comma separated list of up to 10 caller numbers to block from calling this service	String	RW	
MaxSessions	The maximum number of simultaneous calls that may be established on this service	UINT	RW	2
AnonymousCallBlockEnable	Enable blocking of Anonymous Calls on this service. Anonymous calls are rejected with a SIP 486 (Busy) response and Call Forward On Busy service is not applied. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
AnonymousCallEnable	Enable masking of Caller-ID information for all outgoing calls. If enabled, the called party should perceive the call as coming from an anonymous caller. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
DoNotDisturbEnable	Enable Do Not Disturb Service. If enabled, all incoming calls on this service are treated as if the device is busy. Note: It is possible for a user to set this parameter from the phone using a Star Code	Bool	RW	false
OBiTALK Service – Inbound Direct Dialing Authentication (**VoiceService.1.X_P2P.1.VoiceGateway.**)				
AuthMethod	The OBiTALK protocol allows incoming calls to indicate a target number that is different from this device's OBi number. The device in that case will attempt to establish and bridge the call to the target number according to the rules configured in the trunk's InboundCallRoute parameter. Hence this device acts as a gateway and the method is referred to direct dialing or 1-stage dialing (versus 2-stage dialing via the Auto-Attendant). Since the caller is not able to enter a PIN in such cases, an automated method based on signalling protocol must be used to authenticate the caller if authentication is required. OBi device offers the following choices for this purpose: • None = Disable authentication • HTTP Digest = Use HTTP Digest with User-ID and Password pairs. Note that at least one of **AuthPassword**x (x=1,2,3,4) must be specified, otherwise authentication is disabled.	Choice	RW	HTTP Digest
AuthUserID1	One of 4 userids for authenticating direct dialing callers	String	RW	
AuthPassword1	One of 4 passwords for authenticating direct dialing callers	String	RW	
AuthUserID2	One of 4 userids for authenticating direct dialing callers	String	RW	

OBIHAI
technology, inc.

AuthPassword2	One of 4 passwords for authenticating direct dialing callers	String	RW	
AuthUserID3	One of 4 userids for authenticating direct dialing callers	String	RW	
AuthPassword3	One of 4 passwords for authenticating direct dialing callers	String	RW	
AuthUserID4	One of 4 userids for authenticating direct dialing callers	String	RW	
AuthPassword4	One of 4 passwords for authenticating direct dialing callers	String	RW	

Note: If **AuthPassword**x is specified, **AuthUserID**x may be set to blank to let the device use the default value which is a special hash of the **AuthPassword**x. This is only applicable if the external gateway is also an OBi device that understands how to generate the default **AuthUserID**x using the same hash function.

Auto Attendant Service

Automated Attendant

The OBi call processing Auto Attendant (AA) invoked by including "aa" in the inbound call routing rule associated the interface on the OBi processing an incoming call. When connecting to the AA in this manner, there are two options at present.

Note: At present an OBi device supports only 1 session of AA at a time. Additional calls routed to the AA while a session is in progress will be rejected by the AA as busy.

AA Callback Service

The OBi offers two methods for the AA to call you back at a number that you picked (or designated by the admin of the OBi device).

The first method is by statically configuring a trunk's InboundCallRoute. A rule can be added to the InboundCallRoute parameter to have the AA call back the caller's or any other number, if the caller hangs up before the AA answers. The rule should indicate that `aa(callback-number)` is the target destination of the call, where *callback-number* is the number that the AA should call back if the caller hangs up before the AA answers the call. For example, the following rule

`{(<**1>(14089913313|12121559801)):aa($1)}`

says that: if 14089913313 or 12121559801 calls, the call is routed to AA. If caller hangs up before the AA answers, AA calls the number represented by $1. Recall that $1 is expanded into the caller number after processing by the digit map on the left side of the colon. In this case it is the caller's number prepended by **1. The **1 is required for outbound call routing when AA calls back; here it indicates SP1 is to be used for calling back (assuming default value of the AA OutboundCallRoute parameter)

The parameter **AA Service**::**CallbackAnswerDelay** controls the number of milliseconds before AA answers when a callback number is specified as shown in the example. The default value is 10000 ms. Without the *(callback-number)* argument, the AA behaves the normal way and the answer delay is governed by the parameter **AA Service**::**AnswerDelay**.

The second method is by selecting AA option 3 to "Enter a callback number" after the AA answers the call. Caller can explicitly enter the number to be called back by the AA. If a valid number is entered, AA says "Thank You" and "Goodbye", and then starts calling back 2s after the current call has ended. If number entered is invalid, AA plays SIT tone followed by an error message. Note that the variable $1 (representing the caller's number) is carried over to the subsequent AA callback call. The AA DigitMap can include $1 to be used in a callback context. For example, the following rule in the AA DigitMap

`(<00:**1$1>|… )`

says that if the AA dials 00, the device will transform it into the caller's number prepended by **1. In other words, if the caller wants the AA to callback the current number (typically the case), he can simple enter 00# after selecting option 3 on the AA menu. Note that $1 can only be used as part of a substitution element in the digit map; it must not be used for matching elements since its value is unknown.

Automated Attendant:

IVR Announcement	Attendant Announcement	What Happens Next:

Number		
1	Press 1 to continue this call.	When accessed from the OBiTalk,
2	Press 2 to make a new call.	If "UsePIN" authentication is enabled and the user enters a matching PIN, the OBi Attendant will immediately prompt the user to enter number followed by the pound (#) key. If the entered PIN is not a match, the Attendant will give the user two additional attempts to enter the PIN. If the third attempt does not match, the Attendant will announce a thank you message and disconnect the call.
3	Press 3 to enter a callback number.	If a valid number is entered, AA says "Thank you" and "Goodbye", hangs up, and then callback the number in 2s. It the given number is invalid, AA plays SIT tone followed by an error message. Tips: Caller can simply dial 00# to have the AA call back his current number.

User Recorded Prompts

The OBi supports 20 user recordable prompts which are referred to as the *User1* to *User20* prompt respectively. See the section **Telephone-IVR-Based Local Configuration** on how they can be recorded, or the section **Customized AA Prompts Backup & Restore** on how they can be duplicated from one device onto another device.

Customizing AA Prompt Lists

AA does not play individual user prompts directly. Instead it plays a comma separated list of prompt elements, known as a *Prompt List*. A prompt element can be a user prompt with optional parameters, or a control element. A user prompt is referred as %User<N>% where <N> = 1 – 20. In a prompt list this may be followed by a *;r=<start>-<end>* parameter that specifies the range to play for that prompt, where

> <start> = starting time mark in milliseconds; 0 is the default if omitted

> <end> = ending time mark in milliseconds; the end of the prompt is the default if omitted

If the r= parameter is omitted, the full range of the prompt is played.

Examples:

> %User1%;r=1000   = play User1 prompt starting at 1000ms mark to the end

> %User2%   = play the entire User2 prompt from start to finish

> %User3%;r=1300-3720   = play User3 prompt starting from 1300ms mark to the 3720ms mark

> %User4%;r=3200-1200   = does not play anything since <end> is less than <start>

Each prompt list control elements starts with a '&' in a prompt list. The following control elements are supported:

> &pause(<duration>)   = pause playing for a number of seconds as given by the <duration> parameter

An example of prompt list:

%User1%;r=105,&pause(3),%User5%,%User9%;r=0-1350,&pause(15)

You can replace any of the following AA prompt lists with your own specified prompt lists:

AA Prompt List	System Default	Prompt Be Played
Welcome	Welcome to OBi Attendant	Once, at the beginning when the AA starts
InvalidPin	Invalid PIN	After user enters an invalid PIN
EnterPin	Enter PIN	Prompts user to enter a valid PIN
MenuTitle	Main Menu	Once, after Welcome and before announcing the menu options
Menu	Press 1 to continue this call. Press 2 to make a new all. Press 3 to enter a callback number.	A couple of times after MenuTitle
PleaseWait	Please wait while your call is being connected.	Once, after user enters a phone number to call
EnterNumber	Enter number followed by the # key.	Prompts user to enter a valid number after option 2 or option 3 is selected by the user
Bye	Thank you for choosing Obihai Technology. Goodbye.	When user presses * or # key to leave the AA

Auto Attendant Web Page

The following configuration parameters are available on this web page.

Parameter	Description	Type	Access	Default Setting
User Prompts (**VoiceService.1.X_UserPrompt.**)				
User$N$Description N = 1-20	A text string that describes the contents of this user prompt. You can click this parameter to invoke a page to upload an audio file for the prompt (.wav and .au files in 16-bit linear format at 8/11.025/16/22.05/32/44.1/48 kHz sample rate are supported).	String	RW	
User$N$Length N = 1-20	This is a read-only status parameter. It shows the space occupied by this prompt in number of milliseconds	String	R	
EnableDownloadURL	Enable the use of **DownloadURL** to download user prompt package	Bool	RW	false
DownloadURL	A URL to download a user prompt package. WARNING: This will overwrite all existing user prompts)	String	RW	
SpacedUsed	This is a read-only status parameter. It shows the amount of recording space used in number of milliseconds	String	R	
SpaceAvailable	This is a read-only status parameter. It shows the amount of recording space remaining in number of milliseconds	String	R	
Auto Attendant (**VoiceService.1.X_AA.1.**)				
Enable	Enable AA. If enabled, the AA will answer an incoming call that has been routed to it after a period as specified in AnswerDelay. If disabled, the AA will not attempt to answer	Bool	RW	true

OBIHAI
technology, inc.

	any incoming call.			
DigitMap	Once the AA answers an incoming call, it presents the caller with an option to make a further call using one of the available voice services on the device. This Digit map serves to restrict the numbers that can be dialed or called via this AA option. See *OBi Call Routing and Digit Map Section* for a description of the syntaxes to specify a digit map.	String	RW	See insert below.

	Default Setting:
	
	`([1-9]x?*(Mpli)|[1-9]|[1-9][0-9]|<00:$1>|[0-8]|**1(Msp1)|**2(Msp2)|**3(Msp3)`
	`|**4(Msp4)|**81(Mbt)|**82(Mbt2)|**9(Mpp)|(Mpli))`			

OutboundCallRoute	After the caller dials a number that is acceptable by the AA (according to its DigitMap) to make a further call, the device uses this outbound call routing rule to determine which service to make this call with. See *OBi Call Routing and Digit Map Section* for a description of the syntaxes to specify this parameter Note that while forking to multiple numbers in an AA outbound call routing rule is not supported on OBi100 and OBi110, it is supported on the OBi202. For example, on the OBi202 you may have a rule like this: {0:ph,ph2} which forks to ring both PHONE1 and PHONE2. In general you can have up to 4 destinations is a forking rule. On the other hand, you must not specify more than one destination number on the OBi100 and OBi110.	String	RW	See insert below

	Default Setting:
	
	`{([1-9]x?*(Mpli)):pp},`
	`{0|1:ph},{2:ph2},{3:ph3},{4:ph4},{5:ph5},{6:ph6},{7:ph7},{8:ph8},`
	`{(<**1:>(Msp1)):sp1},{(<**2:>(Msp2)):sp2},{(<**3:>(Msp3)):sp3},{(<**4:>(Msp4)):sp4},`
	`{(<**82:>(Mbt2)):bt2},{(<**81:>(Mbt)):bt},{(<**9:>(Mpp)):pp},{(Mpli):pli}`			

OBIHAI
technology, inc.

PrimaryLine	By primary line we mean the service that does not require any access code prefix (such as **1 or **9) when dialing; it is the default service to be used for making the call when no explicit access code prefix is entered. This parameter indicates to the device which voice service is considered as the primary line when dialing out via the Auto Attendant. Available choices are: • `SP1 Service` (code = sp1) • `SP2 Service` (code = sp2) • `SP3 Service` (code = sp3) • `SP4 Service` (code = sp4) • `SP5 Service` (code = sp5) • `SP6 Service` (code = sp6) • `SP7 Service` (code = sp7) • `SP8 Service` (code = sp8) • `SP9 Service` (code = sp8) • `OBiTALK Service` (code = pp1) • `OBiBlueTooth 1` (code = bt1) • `OBiBlueTooth 2` (code = bt2) • `Trunk Group 1` (code=tg1) • `Trunk Group 2` (code=tg2) The OBi device process the parameter by substituting of the occurrences of *pli* and *(Mpli)* in DigitMap and OutboundCallRoute with the corresponding *code* and *(Mcode)*.			`SP1 Service`
AnswerDelay	Period of time in milliseconds that the AA will wait before answering an incoming call that has been routed to it			`4000`
NumberOnNoInput	In the case that the caller does not enter any option from the top level menu after the menu has been announced for 3 times, the AA directs the caller to the number specified in this parameter. If this number is not specified, the AA simply terminates the current call. Note: According to the default **DigitMap** and **OutboundCallRoute**, calling 0 or 1 rings PHONE1, 2 rings PHONE2, 3 rings PHONE3, etc.			`0`
UsePIN	Enable the use of PIN to authenticate callers when they select the option to make a further call. If PIN1, PIN2, PIN3, and PIN4 are all empty, device treats it as if UsePIN is set to No. Otherwise, the caller must enter one of the non-empty PIN in order to proceed,			`false`
PIN1	PIN code to make a call (must be all digits). Maximum Length = 15			
PIN2	PIN code to make a call (must be all digits). Maximum Length = 15			
PIN3	PIN code to make a call (must be all digits). Maximum Length = 15			
PIN4	PIN code to make a call (must be all digits). Maximum Length = 15			
Auto Attendant Prompts (VoiceService.1.X_AA.1.Prompt.)				
Welcome	Prompt List to replace the system's Welcome message. You may click this parameter to play it from the web browser	String	RW	

OBIHAI
technology, inc.

InvalidPin	Prompt List to replace the system's InvalidPin message. You may click this parameter to play it from the web browser	String	RW	
EnterPin	Prompt List to replace the system's EnterPin message. You may click this parameter to play it from the web browser	String	RW	
MenuTitle	Prompt List to replace the system's MenuTitle message. You may click this parameter to play it from the web browser	String	RW	
Menu	Prompt List to replace the system's Menu message. You may click this parameter to play it from the web browser	String	RW	
PleaseWait	Prompt List to replace the system's PleaseWait message. You may click this parameter to play it from the web browser	String	RW	
EnterNumber	Prompt List to replace the system's EnterNumber message. You may click this parameter to play it from the web browser	String	RW	
Bye	Prompt List to replace the system's Bye message. You may click this parameter to play it from the web browser	String	RW	
Prompt-Map-Route 1 (VoiceService.1.X_PMR.1.)				
PromptList	See the *Prompt List* subsection in the *Multi-Level AA with PMR* section.	String	RW	`music`
DigitMap	A digit map that governs what digit patterns can be entered while this prompt list is being played	String	RW	
OutboundCallRoute	An outbound call route that controls how to route the call based on the digits entered	String	RW	
Prompt-Map-Route 2 (VoiceService.1.X_PMR.2.)				
PromptList	See the *Prompt List* subsection in the *Multi-Level AA with PMR* section. Default value: `'Thank you' for 'your call',&pause(1)`	String	RW	See desc.
DigitMap	A digit map that governs what digit patterns can be entered while this prompt list is being played	String	RW	`([0-8])`
OutboundCallRoute	An outbound call route that controls how to route the call based on the digits entered Default value: `{(x?):pm(3;n=2;i=$2)}`	String	RW	See desc.
Prompt-Map-Route 3 (VoiceService.1.X_PMR.3.)				
PromptList	See the *Prompt List* subsection in the *Multi-Level AA with PMR* section. Default Value: `Please enter extension,&pause(5)`	String	RW	See desc.
DigitMap	A digit map that governs what digit patterns can be entered while this prompt list is being played	String	RW	`([0-8])`

OutboundCallRoute	An outbound call route that controls how to route the call based on the digits entered
	
	Default value:
	`{?|0|1:ph},{2:ph2},{3:ph3},{4:ph4},{5:ph5},`
	`{6:ph6},{7:ph7},{8:ph8}`
		String	RW	See desc.

Prompt-Map-Route *n* (VoiceService.1.X_PMR.*n*.) for *n* = 4, 5, …, 12				
PromptList	See the *Prompt List* subsection in the *Multi-Level AA with PMR* section	String	RW	

DigitMap	A digit map that governs what digit patterns can be entered while this prompt list is being played	String	RW	
OutboundCallRoute	An outbound call route that controls how to route the call based on the digits entered	String	RW	

Call Park

Call park is a special terminal that lets a phone user park the call and later retrieve it. Call park is referred to with the short name in PK in call routing rules. You can specify some parameters in parenthesis when using PK in a routing rule. The general syntax is

pk(*type;*s=*source*[;d=*delay*][;n=*iteration*])

where

type = one of the following (required)

- park*xx* (literal 'park' followed by 2-digit slot ID *xx*) – Park a call at slot *xx*

- pickup*xx* (literal 'pickup' followed by 2-digit slot ID *xx*) – Pickup the call at slot *xx*

- aa – Use the parking lot as the entry point to a multi-level AA based on PMR*

- moh – Use the parking lot as an MOH Source

source = ID (1–12) of the PMR* to use (required)

delay = Delay in seconds in answering (optional)

iteration = Number of times to play the prompt (in the given PMR); 0 means play forever. Default is 0. (optional)

* See the section ***Multi-Level AA with PMR*** for more details on PMR

Examples:

– pk(aa;d=4;s=2;n=1)

– pk(moh;s=1)

By default factory configuration, a phone user can dial 6xx to park a call at the slot xx and dial 7xx to pick up the call parked at slot xx (where xx is a 2-digit number). User may dial 6** to park the call at the first available slot in the range 00 – 99, and 7** to pick up to oldest call parked in the range 00 – 99. Similarly, user may dial 6x* to park the call in the first available slot in the range x0 – x9 and dial 7x* to pick up the oldest call parked in the range x0 – x9 (where x is a digit 0–9). Whenever one or more call is parked (with type = park*xx*) , all of the phones will play a short ring periodically as a reminder. The pattern of this ring is configurable using the parameter X_CallParkedRing on the SP*n* Service Web Page.

If the user attempts to park a call at a slot that has already been occupied, she will hear fast busy. If the user attempts to pick up a call from a slot that is unoccupied, she will hear fast busy also. Calls parked as type aa or moh are not available for call pickup.

Multi-Level AA with PMR

The AA described above provides a simple single-level menu. In this section we describe another method that can be used to provide a more general multi-level AA function. This method is based on PMR, which stands for **P**rompt-digit**M**ap-call**R**oute.

Prompt List

A Prompt List is A comma separated list of *prompt elements* and *control elements*, such as such as:

```
'Thank you' for 'your call', &pause(1)
```

A prompt element in a prompt list can be a system provided prompt or a user recorded prompt. A system prompt is referred to by its name in the prompt list, such as *music*, while a user prompt is referred to by its name surrounded by a pair of %, such as *%USER1%*, *%USER2%*, etc. Each prompt element may include an "r=" parameter that specifies the range to play for that prompt. The full syntax of a prompt element is:

<Prompt Name>[;r=[<start>][-<end>]]

where

- [x] indicates an optional element x

- <Prompt Name> = Name of the prompt. Currently available are *%User1%, …, %User20%*, and

    o `music` (a system provided music fragment)

    o `'Thank you' for 'your call'` (the phrase: Thank you for your call)

    o `Please enter extension` (the phrase: Please enter extension)

- <start> = starting point in milliseconds to play the given prompt; play from the beginning if this value equals to 0 or not specified

- <end> = ending point in milliseconds to play the given prompt; play to the end if this value equals to 0 or not specified

- If the [;r=…] parameter is not specified, play the entire prompt from the beginning to the end

Examples of prompt element:

- %User1%   (play the User1 prompt from the beginning to the end)

- music;r=1000-8500   (play the music prompt starting from the 1000ms mark to the 8500ms mark

- %User2%;r=-2500   (play the User2 prompt starting from beginning to the 2500ms mark

- %User20%;r=1300   (play the User20 prompt starting from the 1300ms mark to the end

- %User15%;r=32-12   (play anything since <end> < <start>)

A control element in a prompt list starts with the letter '&'. The following control elements are supported:

- &pause(<duration>) – Pause for the given duration in seconds. For example: &pause(10)

- &call([target]) – Call the target number given in trunk(number) format. For example: &call(sp1(14083331234)), &call(aa3)

PMR

A PMR is a triplet parameter object that includes:

- A Prompt List as described in the last section

- A DigitMap: Acceptable digit patterns while the prompt list is being played

- A CallRoute: Where to send the call when a matching digit pattern is entered.

You can define up to 12 PMR in the OBi500, where each is identified by the ID 1 – 12. The PMR parameters are grouped under the Auto Attendant web page. A PMR is invoked by specifying its ID as the value of the "*s*" parameter in the argument of a PK (Call Park) Terminal in a call routing rule. For example:

pk(aa;d=4;s=2;n=1)

routes the call to the parking lot that

- Behaves as an AA

- Answers the call in 4 seconds

- Starts with PMR 2 and plays the prompts list once

Rules in a PMR's CallRoute parameter can be specified to refer to another PMR with the general syntax:

pm(*id*[;n=*iteration*][;i=*input*])

where

- *id*: ID of the target PMR to "jump to"; -1 will "go-back" to the last PMR in the stack (or end the call if already at the top of the stack

- *iteration*: number of times to play the prompt list for; 0 to play forever (optional with 0 implied)

- *input*: optional pre-entered input digits, if any, to pass to the target PMR; this allows user to dial-ahead during the current PMR the input for the next PMR. You can set *input* = $2 to use the matching input digits (after transformation by the DigitMap) that triggers the rule in CallRoute as the pre-entered input. For example:

{(x?):pm(6;n=2;i=$2)}

Multiple PMR can be daisy-chained to support the function of a multi-level AA. By default, PMR1, PMR2, and PMR3 are configured with the following values:

- PMR 1 Default:

    – PromptList = music

    – DigitMap = <blank>

    – OutboundCallRoute = <blank>

    – Suitable to be used for MOH

- PMR 2 Default:

    – PromptList = 'Thank you' for 'your call',&pause(1)

    – DigitMap = ([0-8])

- OutboundCallRoute = {(x?):pm(3;n=2;i=$2)}

- PMR 3 Default:

  - PromptList = Please Enter Extension,&pause(5)

  - DigitMap = ([0-8])

  - OutboundCallRoute = {?|0|1:ph} ,{2,ph2} ,{3,ph3} ,{4,ph4} ,{5,ph5} ,{6,ph6} ,{7,ph7} ,{8,ph8}

Default PMR 2 and PMR 3 form a simple AA as found in OBiPLUS default AA3/AA4. To use this function, simply route an incoming call to pm(2)

Voice Gateways and Trunk Groups

Voice Gateway

A gateway in this context is another OBi device which lets incoming OBiTALK callers to call further on one or more of its trunks (such as SP1). The caller can call the gateway first with a normal OBiTALK call, get the AA, and then dial the target number. For authentication the AA may ask the user to enter a PIN before establishing the second call. This way of dialing is known as 2-stage dialing.

On the other hand, a gateway can be configured on the originating OBi device such that the caller can dial the target number directly without going through the AA. We refer to this method of dialing as direct dialing or 1-stage dialing. Since it is not possible to enter a PIN in the case of direct dialing, a userid/password pair can be configured for the gateway also so that the device can authenticate with the gateway automatically using HTTP digest method. HTTP digest authentication is optional. You do not need to provide user/password if the gateway does not require authentication for direct dialing.

OBi allows the user to specify up to 8 gateways. Each gateway is addressed using its factory-assigned OBi Number. A gateway is conceptually a trunk with its own DigitMap. You can refer to a gateway and its associated DigitMap with the short trunk name VGn and (Mvg$n$) respectively, for n = 1, 2, 3, …, 8. VGn and (Mvg$n$) can be used in call routing rules and digit maps just like other real trunks.

As an example, you can add the rule {(1xxx xxx xxxx):vg2} in PHONE port's OutboundCallRoute to let the device dials out using VGs when caller dials any 11-digit number starting with 1. On the gateway side, you can add the corresponding rule {>(1 xxx xxx xxxx):sp1} in the OBiTALK Service::InboundCallRoute to make the call on its SP1 trunk. You can change the last rule to {(290 333 100|200 444 101)>(1 xxx xxx xxxx):sp1} if you want to limit the gateway to allow just the two stated caller numbers to make such calls.

A gateway may also be configured with a SIP URL as the access number to be accessed by the device over one of the SP trunks. For example, one can set the gateway access number as SP1(some-sip-server.mydomain.com), or SP2(192.168.15.111:5062), etc. Note that when using a SP trunk to access a (SIP) gateway, the device will:

- Not use the outbound proxy, ICE, or STUN regardless the settings on the SP trunk.

- Use only the device's local address as the SIP Contact, and ignore any natted address discovered by the device.

- Use the gateway's SIP URL to form the FROM header of the outbound INVITE.

- Use the gateway's AuthUserID and AuthPassword for authentication.

- Apply the symmetric RTP concept.

Trunk Groups

As the name implies, a trunk group is a group of trunks. If a call is routed to a trunk group, OBi picks one of the available trunks from the group to make the call. Availability of trunk is based on:
- Whether the trunk's digit map allows the number to call, AND
- Whether the trunk has capacity to make one more call

Up to 4 trunk groups can be configured on an OBi device. Each trunk group is conceptually another trunk with its own DigitMap. A trunk group and its associated DigitMap are referenced using the short name TGn and (Mtg$n$) respectively, where n = 1, 2, 3, 4. They can be referenced in other digit maps and call routing rules so that calls may be routed to a particular trunk group.

Only trunks can be added to a trunk group. These include: PP1, SP1 – SP9, BT1, BT2, VG1, VG2, …, VG8, TG1, TG2, … TG4. Note that a TG may include another TG (that is, TG can be recursive). However, you must make sure this does not result in infinite recursion.

Gateways and Trunk Groups Web Page

Parameter	Description	Type	Access	Default Setting
Voice Gateway *n* (**VoiceService.1.X_VoiceGateway.*n*.**) for *n* = 1–8				
Enable	Enable this voice gateway	Bool	RW	`true`
Name	An arbitrary user-friendly name to identify this gateway (optional)	String	RW	
AccessNumber	The gateway's OBiTALK number, including trunk information, such as: `PP(ob200112334)` or `PP(ob300331456)` If the value is *blank*, device treats this VG as disabled. Starting with release 1.2, this can also be set to a SIP URL, such as: `SP1(sip.mycompany.com:5060)`, or `SP2(192.168.15.113)`	String	RW	
DigitMap	Digit Map for this VG. It can be referenced as `(Mvgn)`	String	RW	`(xx.)`
AuthUserID	A User-ID to authenticate with the gateway	String	RW	
AuthPassword	A Password to authenticate with the gateway	String	RW	
Trunk Group *n* (**VoiceService.1.X_TrunkGroup.*n*.**) for *n* = 1–4				
Enable	Enable this trunk group	Bool	RW	`true`
Name	An arbitrary user friendly name to identify this trunk group (optional)	String	RW	
TrunkList	A comma separated list of names of trunks to include in this trunk group. Default Setting: `sp1,sp2,sp3,sp4`　(for *n* = 1, 2) *blank*　(for *n* = 3, 4)	String	RW	See desc.
DigitMap	Digit map associated with this trunk group. It can be referenced as (Mtg*n*) Default Setting: `(Msp1)`　(for *n* = 1, 2) `(xx.)`　(for *n* = 3, 4)	String	RW	See desc.

OBiBlueTooth and OBiBT

OBiBlueTooth *n* Web Page

The configuration parameters on this page are listed below.

Parameter	Description	Type	Access	Default Setting
OBiBlueTooth *n* (**VoiceService.1.X_BT.*n*.**) for *n* = 1, 2				
Enable	Enable this OBiBlueTooth Service	Bool	RW	true
DigitMap	Digit Map associated with this service. It can be referred as (Mbt*n*) in other digit maps and call routing rules Default Setting: `([2-9]xxxxxxS4|1xxxxxxxxxx|011xx.|[1-9]11S2|[1-9]xx)`	String	RW	See Desc.
InboundCallRoute	A set of call routing rules to control how to route incoming call on this service	String	RW	ph,ph2
RingProfile	The Ring profile to use to ring a phone port for incoming calls on this service. Choices are: A, B	Choice	RW	A
DefaultRing	Default ring pattern to use to ring a phone port for incoming calls on this service. Choices are: 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	1
CallOnHoldRing	The ring pattern to use to ring a phone port to remind holding call on this service. Choices are: No Ring, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10	Choice	RW	8
DirectoryNumber	Informational only. The PSTN number associated with this service	String	RW	
OBiBlueTooth *n* – Calling Features (**VoiceService.1.X_BT.*n*.CallingFeatures.**) for *n* = 1, 2				
CallForwardUnconditionalEnable	Enable call forward unconditionally by the device to all incoming calls on this trunk	Bool	RW	false
CallForwardUnconditionalNumber	The number to forward all incoming calls on this trunk to unconditionally	String	RW	
CallForwardOnBusyEnable	Enable call forward on busy by the device to incoming calls on this trunk	Bool	RW	false
CallForwardOnBusyNumber	The number to forward incoming calls on this trunk to when busy	String	RW	
CallForwardOnNoAnswerEnable	Enable call forward on no answer by the device to incoming calls on this peripheral	Bool	RW	false
CallForwardOnNoAnswerNumber	The number to forward incoming calls on this trunk to when not answered within a given number of rings	String	RW	
CallForwardOnNoAnswerRingCount	The number of rings before triggering call forward no answer by the device to incoming calls on the trunk. One ring is equivalent to 6s.	UINT	RW	2
BlockedCallers	A comma separated list of up to 10 caller numbers to block from calling this service	String	RW	
AnonymousCallBlockEnable	Enable Anonymous Call Blocking feature to incoming calls on this trunk	Bool	RW	false
DoNotDisturbEnable	Enable Do Not Disturb feature to incoming calls on this trunk	Bool	RW	false

BridgedOutboundCallMaxDuration	Limit on the duration of all bridged outbound calls on this trunk. Unit is in seconds. *Blank* or `0` implies call duration is not limited	UINT	RW	
OBiBlueTooth *n* **– Device Settings** (**DeviceInfo.Bluetooth.Basic.** for *n* = 1, **DeviceInfo.Bluetooth.Basic.2.** for *n* = 2) for *n* = 1, 2				
Discoverable	A status that indicates if the OBiBT on this channel is currently discoverable	Bool	R	`false`
PreferredPairedDevice	Select the preferred BT device when more than one paired devices are in range. Choices are: • `None` • `Device 1` • `Device 2` • `Device 3` • `Device 4` • `Device 5` • `Device 6` • `Device 7` • `Device 8` • `Device 9` • `Device 10`	Choice	RW	`None`
PairedDevice*N* Repeat for *N* = 1 – 10	The name of an external paired BT device.	String	R	
RemovePairedDevice*N* Repeat for *N* = 1 – 10	Check this box and press submit on the web page to remove this paired device	Bool	RW	`false`

Phone Interface Features of the OBi Device

Repeat Dialing Service

Repeat dialing service is when a user dials *05 to tell the device to redial the last called number repeatedly while the phone is onhook, until the called party rings or answers. When that happens, device rings the PHONE port and the user can pick it up to talk to the called party. Typically the last called number was busy when the user invokes this feature, but the device allows this feature for all cases.

This feature can be controlled with the following two parameters (under the PHONE Port – Calling Feature section):

- RepeatDialInterval = the minimum number of seconds between each redial. Default is 30s

- RepeatDialExpires = the maximum duration in seconds when the repeat dialing remains active. Default is 1800s

User dials *06 to cancel Repeat Dialing. Only one repeat dial request is supported. Dialing *05 while a repeat dial is in progress will be rejected with a fast busy tone. If *05 is accepted, the device plays normal dial tone.

Notes:

- The first redial happens 5s after the phone is onhook following *05

- When phone is offhook or rings for an incoming call, device pauses redial and cancels the call if it's already dialed but the peer device is not ringing yet.

- As soon as phone goes on hook or ringing stops without any calls on hold, repeat dialing resumes in 5s

- If called party answers before the local caller, device sends normal ringback tone over RTP to the called party

- The ring for alerting local user when the called party rings or answers is taken from outgoing trunk's RepeatDialRing parameter

- Repeat Dial service cannot be used on the LINE port. If the last call was made over the LINE port, dialing *05 will result in fast busy

- Repeat Dial calls are not logged to call history, except the last and successful one when the called party rings or answers

PHONE*n* Port Web Page (*n* = 1, 2, 3, 4, 5, 6, 7, 8)

Parameter	Description	Type	Access	Default Setting
PHONE*n* Port – Phone Port (**VoiceService.1.X_FXS.***n***.**) for *n* = 1, 2, …, 8				
Enable	Enable the PHONE port	Bool	RW	true
DigitMap	This Digit map serves to restrict the numbers that can be dialed or called from the PHONE port. If the caller dials a number that is not allowed by the digit map, OBi plays SIT tone followed by a short error message to let the caller know that the dialed number is invalid. See *OBi Call Routing and Digit Map Section* for a description of the syntaxes to specify a digit map.	String	RW	See insert below

	Default Setting: `([1-9]x?*(Mpli)|[1-9]S9|[1-9][0-9]S9|911|[67]XX|**0|***|#S4|#[0-8]|#9x| **81(Mbt)|**82(Mbt2)|**1(Msp1)|**2(Msp2)|**3(Msp3)|**4(Msp4)|**9(Mpp)|(Mpli))`			
OutboundCallRoute	After the caller dials a number that is acceptable according to the DigitMap, OBi device uses this outbound call routing rule to determine which service to make this call with. If no appropriate call route found, OBi plays SIT tone followed by a short error message to let the caller know that there is no call route to place the call. See *OBi Call Routing and Digit Map Section* for a description of the syntaxes to specify this parameter	String	RW	See insert below
	Default Setting: `{([1-9]x?*(Mpli)):pp},{#0:ao},{#1:ph1},{#2:ph2},{#3:ph3},{#4:ph4},{#5:ph5}, {#6:ph6},{#7:ph7},{#8:ph8},{#*:ph1,ph2,ph3,ph4,ph5,ph6,ph7,ph8}, {(<6:park>XX<;s=1>):pk},{(<7:pickup>XX<;d=0>):pk}, {(<**82:>(Mbt2)):bt2},{(<**81:>(Mbt)):bt},{**0:aa},{***:aa2}, {(<**1:>(Msp1)):sp1},{(<**2:>(Msp2)):sp2},{(<**3:>(Msp3)):sp3}, {(<**4:>(Msp4)):sp4},{(<**9:>(Mpp)):pp},{(Mpli):pli}`			
CallReturnDigitMaps	Call Return is the service where the user can call the last caller by dialing a star code (*69 by default). OBi device implements this service by remembering the number of the last caller in memory. However the stored information does not include any dialing prefix to tell the device which voice service to use to call back the last caller. This list of digit maps serve the purpose of mapping a caller's number to one that includes the desired dialing prefix used exclusively for call return service.	String	RW	See insert below
	Default Setting: `{pli:(xx.)},{sp1:(<**1>xx.)},{sp2:(<**2>xx.)}, {sp3:(<**3>xx.)}, {sp4:(<**4>xx.)},{bt1:(<**81>xx.)},{bt2:(<**82>xx.)},{pp:(<**9>xx.)}`			

PrimaryLine	By primary line we mean the service that does not require any access code prefix (such as **1 or **9) when dialing; it is the default service to be used for making the call when no explicit access code prefix is entered. This parameter indicates to the device which voice service is considered as the primary line when dialing out from the PHONE port. Available choices are: `SP1 Service` (code = sp1) `SP2 Service` (code = sp2) `SP3 Service` (code = sp3) `SP4 Service` (code = sp4) `SP5 Service` (code = sp5) `SP6 Service` (code = sp6) `SP7 Service` (code = sp7) `SP8 Service` (code = sp8) `SP9 Service` (code = sp9) `OBiTALK Service` (code = pp1) `OBiBlueTooth 1` (code=bt1) `OBiBlueTooth 2` (code=bt2) `Trunk Group 1` (code=tg1) `Trunk Group 2` (code=tg2) The OBi device process the parameter by substituting of the occurrences of `pli` and `(Mpli)` in **DigitMap**, **OutboundCallRoute**, and **CallReturnDigitMaps** with the corresponding *code* and `(Mcode)`.	Choice	RW	SP1 Service
ToneOnPrimaryServiceDown	Multiple Choices: `No Tone`: `Normal Dial Tone`: `SIT Tone 1`: `SIT Tone 2`: `SIT Tone 3`: `SIT Tone 4`:			Normal Dial Tone
PHONE*n* Port – Ringer (**VoiceService.1.X_FXS.*n*.Ringer.**) for *n* = 1, 2, …, 8				
RingFrequency	Ringer frequency in Hz (14 - 68) to apply to the PHONE port when ringing	UINT	RW	20
RingVoltage	Peak ringer voltage in volts (55 - 82) to apply to the PHONE port when ringing	UINT	RW	70
RingWaveform	Ringer waveform to apply to the PHONE port when ringing. Choices are `Sinusoidal`, `Trapezoidal`	Choice	RW	Sinusoidal
PHONE*n* Port – Port Settings (**VoiceService.1.X_FXS.*n*.Settings.**) for *n* = 1, 2, …, 8				
OnHookTipRingVoltage	Tip/Ring Voltage when the attached phone is on hook (30 v to 52 v)	UINT	RW	46
OffHookCurrentMax	Maximum supported current (15 mA to 45 mA) when the attached phone is off-hook	UINT	RW	20
Impedance	PHONE port impedance setting. Available choices are (units in ohm if not specified): `600`	Choice	RW	600

OBIHAI
technology, inc.

	`900` `270+(750||150 nF)` `220+(820||120 nF)` `370+(620||310 nF)` `320+(1050||230 nF)` `350+(1000||210 nF)` `200+(680||100 nF)` `600+2.16 uF` `900+2.16 uF` `600+1 uF` `220+(820||115 nF)`			
OnHookPowerSaveMode	Enable Power Save Mode in on-hook state. This is a global setting and is applied to all phone ports. It is only available in **Phone1 Port**.	Bool	RW	`true`
DTMFPlaybackLevel	Out of band DTMF tone playback level in dBm (-90 to 3)	INT	RW	`-15`
CallerIDMethod	Caller ID delivery standard. Choices are: `FSK(Bell202)` `FSK(V.23)` `DTMF(Finland,Sweden)` `DTMF(Denmark)` `DTMF(Netherlands)` `NTT(Japan)`	Choice	RW	`FSK(Bell202)`
CallerIDTrigger	Triggering event for on-hook Caller ID signal generation. Choices are: `After First Ring`: `After Polarity Reversal`: `Before First Ring`:	Choice	RW	`After First Ring`
ChannelTxGain	Transmit gain in dB (-12 to 12) to apply to signal sent from OBi to the attached phone(s)	INT	RW	`0`
ChannelRxGain	Receive gain in dB (-12 to 12) to apply to signal received by OBi from the attached phone(s)	INT	RW	`0`
SilenceDetectSensitivity	PHONE port silence detection servers the purpose of driving silence suppression in RTP transmission when the phone Call terminates on SP1/2 or OBiTALK Service and silence suppression is enabled. This parameter is used to set a sensitivity level for OBi silence detection algorithm. Available choices are: `Low` (harder to detect silence) `Medium` (suggested) `High` (easier to detect silence)	Choice	RW	`Medium`
PHONE*n* Port – Calling Features (**VoiceService.1.X_FXS.*n*.CallingFeatures.**) for *n* = 1, 2, …, 8				
InternalCallerIDName	The Caller ID Name to use when calling Phone-to-Phone	String	RW	`Phone `*n*
InternalCallerIDNumber	The Caller ID Number to use when calling Phone-to-Phone	String	RW	`#`*n*

OBIHAI technology, inc.

CallCommandSignalMethod	Select the method to signal a command to the OBi when the phone is off-hook with an active call in connected state, while there is a second call on hold or ringing. The two choices are: • N. America (uses hook switch events only) • Nordic Regions (R1, R2, …), where R = hook flash or the 'R' button, R0 = Reject the 2nd incoming call (applicable only if the 2nd call is ringing), R1 = End current call, resume/answer the 2nd call R2 = Hold current call, resume/answer the 2nd call R3 = Conference the two calls R4 = Transfer 2nd call peer to the 1st (not applicable if 2nd call is ringing)	Choice	RW	N. America
CallerIDEnable	Enable Caller ID Signal generation. This option can be set to Yes even if the attached phone is not capable of displaying Caller ID. There is no harm in sending Caller ID signal while the phone is in the on hook state.	Bool	RW	false
CallWaitingCallerIDEnable	Enable Call Waiting Caller ID (CWCID) Signal generation. CWCID signal is sent to the phone when it is in the off hook state. It starts with a handshake between the OBi device and the attached phone, by exchanging audible short tones. OBi will proceed with the transmission of the remaining Caller ID signal only if the handshake succeeds (with a phone is capable of displaying CWCID). In that case the phone mutes the handset earpiece until the CWCID signal is complete. Some users however may still find the audible handshake tones objectionable, especially if their phones do not support CWCID. We recommend to those users to set this option to No, if they do not want CWCID feature, or do not have phones that are capable of displaying CWCID.	Bool	RW	false
MWIEnable	Enable MWI Signal (stutter dial tone) generation. If enabled, any SP voice service enabled on the device that has MWI Service enabled will trigger the generation of stutter dial tone if there are new voicemails for the subscriber on the service provider's voicemail system.	Bool	RW	true
VMWIEnable	Enable VMWI Signal generation. If enabled, any SP voice service enabled on the device that has VMWI Service enabled will trigger the generation of VMWI signal if there are new voicemails for the subscriber on the service provider's voicemail system.	Bool	RW	true
CallTransferEnable	Enable Call Transfer. Call Transfer, if enabled, is initiated by the user by hanging up the phone in one of the following scenarios: - One call on hold while a 2nd outgoing call ringing - One call on hold while a 2nd outgoing call connected - One call connected while a 2nd outgoing call ringing - 3-way conference with both calls connected If Call Transfer is disabled, hanging up the phone in the above scenarios simply ends all the calls, except for the one that is holding, which will remain on hold (cases 1 and 2).	Bool	RW	true
ConferenceCallEnable	Enable 3-way Conference Call w/ local audio mixing. Conference Call, if enabled, is initiated by the user by hook	Bool	RW	true

	flashing the phone in one of the following scenarios: - One call on hold while a 2nd outgoing call ringing - One call on hold while a 2nd outgoing call connected We refer to case (1) as an early conference, where the second conferee is still ringing; the other 2 parties may converse while hearing ringback tone in the background until the 3 party answers. In either case, the user can end the call with the second conferee by hook flashing another time and the call reverts to a 2-way call. If Conference Call service is disabled, then hook flashing the phone resumes the holding call but ends the second outgoing call in scenario (1), and swaps between the two calls in scenario (2) (as in a call waiting situation)			
UseExternalConferenceBridge	Use external conference bridge when starting a conference call	Bool	RW	false
StartConfOnPeerRing	Allow the start of a 3-way conference call when the 3rd party is still ringing	Bool	RW	true
CallWaitingEnable	Enable call waiting service. Call Waiting is the situation where a new incoming call is routed to the PHONE port when there is already another call connected. If this service is enabled, OBi plays call-waiting tone to alert the user, as well as generates CWCID signal if CWCID is enabled. The user may then swap between the two calls by hook flashing. If the service is disabled, OBi rejects the incoming call as busy. Note: It is possible for the user to set this parameter from the phone using a Star Code	Bool	RW	true
DoNotDisturbEnable	Enable Do Not Disturb feature on this phone port such that all calls that try to ring this phone port are treated as busy	Bool	RW	false
ToneProfile	Select a Tone Profile for call progress tone generation. Choices are A, B	Choice	RW	A
StarCodeProfile	Select a Star Code Profile for interpreting Star Codes entered by the user. Choices are None, A, B, C, D, E, F, G, or H. If value is set to None, no star code will be recognized by OBi device.	Choice	RW	A
LastDialedNumber	Last number dialed out on the PHONE port.	String	R	
LastCallerNumber	Last caller's number that rings the PHONE port.	String	R	
AcceptMediaLoopback	Enable the device to accept incoming media loopback calls	Bool	RW	true
MediaLoopbackAnswerDelay	Delay in milliseconds before the device answers an incoming media loopback call	UINT	RW	0
MediaLoopbackMaxDuration	Maximum duration in seconds to allow for an inbound media loopback call. Set the value to blank or 0 to make it unlimited	UINT	RW	0
RepeatDialInterval	Interval in seconds between redial in a repeat dial operation.	UINT	RW	30
RepeatDialExpires	Duration of time in seconds when a repeat dial operation remains active.	UINT	RW	1800

OBIHAI
technology, inc.

GenerateCPCSignal	Controls when the device should generate a CPC signal when the remote party hangs up on an established call. The choices are: • `Never` • `For Inbound Calls Only` • `For Outbound Calls Only` • `For Inbound and Outbound Calls`			`For Inbound and Outbound Calls`
UseForPagingOnly	Enables the OBi to be used for paging only when the PHONE port is connected to an external PA system (via a RJ11 to line out connector, available from many electronics shops). In such configuration the PHONE port is expected to be "off-hook" all the time; the OBi will automatically answer an incoming call and will not accept call-waiting.	Bool	RW	`false`
UseForFaxOnly	Assume all calls are for FAXing (not relying on FAX tone detection)	Bool	RW	`false`
FaxDetectionMethod	Choices are: • `CNG or CED` • `CNG only` • `CNG or V.21 preamble`			`CNG or CED`
TransferWhenHolding	This option provides a short cut to transfer a call to a fixed pre-configured number without dialing it. If a valid number is specified for this parameter, the OBi will transfer the call to the given number when the phone hook flashes and then on-hook (which would normally leave the call holding if this parameter is not specified). The valid number should be a complete number with trunk information, such as `SP1(14083334567)`.	String	RW	
EndHoldingCallWhenHangUp	Enable hanging up of holding calls when the local phone is onhook; otherwise the OBi plays a short hold-reminder-tone to indicate there are still calls on hold	Bool	RW	`false`
EndConfWhenHangUp	End 3-way (local mixing) conference call when hang up, instead of transferring one remote party to the other	Bool	RW	`false`
MOHServiceNumber	A number to call and bridge to get some audio played when the phone user holds the call	String	RW	`pk(moh; s=1)`
PlaySITOnCallFailureCodes	Specify which INVITE failure response codes will trigger the OBi to play SIT tone to alert the phone user. It must be specified with a valid digitmap format. If nothing valid is specified, the OBi plays normal reorder tone when the outbound call fails.	String	RW	`([4-9]xx)`
PlaySITWithAnnouncement	Enable whether the OBi should play an announcement of the error code after SIT tone is played when an outbound calls fails. Announcement will not be played if SIT is not played.	Bool	RW	`false`
PHONE*n* Port – Call Forwarding (VoiceService.1.X_FXS.*n*.CallForward.) for *n* = 1, 2, …, 8				
ForwardAll	Enable call forward all incoming calls that try to ring this	Bool	RW	`false`

OBIHAI
technology, inc.

	phone port unconditionally			
ForwardAllNumber	The number to forward incoming calls to when **ForwardAll** is enabled and triggered	String	RW	
ForwardOnBusy	Enable call forward incoming calls that try to ring this phone port when it is busy	Bool	RW	`false`
ForwardOnBusyNumber	The number to forward incoming calls to when ForwardOnBusy is enabled and triggered	String	RW	
ForwardOnNoAnswer	Enable call forward incoming calls that ring this phone port but not answered within a given number of rings	Bool	RW	`false`
ForwardOnNoAnswerNumber	The number to forward incoming calls to when **ForwardOnNoAnswer** is enabled and triggered	String	RW	
ForwardOnNoAnswerRingCount	Number of rings on this phone port before triggering **FowardOnNoAnswer**	UINT	RW	`4`
PHONE*n* Port – Timers (**VoiceService.1.X_FXS.*n*.Timer.**) for *n* = 1, 2, …, 8				
HookFlashTimeMax	Hook Flash is a quick transition of the phone's hook switch from Off-Hook state to On-Hook state, and back to Off-Hook state. This parameter specifies the upper time limit in milliseconds such that if the hook switch stays at the intermediate On-Hook state for longer than this time limit, the OBi device will not recognize the state transition as a HOOK FLASH event, but instead as an ON HOOK event followed by an OFF HOOK event	UINT	RW	`900`
HookFlashTimeMin	Hook Flash is a quick transition of the phone's hook switch from Off-Hook state to On-Hook state, and back to Off-Hook state. This parameter specifies the lower time limit in milliseconds such that if the hook switch stays at the intermediate On-Hook state for less than this time limit, the OBi device will not recognize the state transition as a HOOK FLASH event, but consider the hook switch remains at Off-Hook state throughout the transition (in other words, the transition is discarded as a glitch if it happens too quickly)	UINT	RW	`70`
ReorderDelayTime	Delay in ms to start reorder tone after peer hangs up	UINT	RW	`5500`
CPCDelayTime	A short delay in milliseconds before OBi generates a CPC signal to the PHONE port after the far end has hung up during a call	UINT	RW	`2000`
CPCDuration	OBi device generates CPC (Calling Party Control) Signal by removing power from the PHONE port for a short period. This parameter specifies the length of this period in milliseconds. CPC signal tells the attached phone equipment that the far end has ended the call	UINT	RW	`500`
DigitMapLongTimer	Default number of seconds before the digitmap processor times out waiting for more digits for a match or declares a matched pattern with an unspecified length (as in xx. for example) a win	UINT	RW	`10`
DigitMapShortTimer	Default number of seconds before the digitmap processor times out to declare a fully-matched candidate a win while waiting for more digits for other potential matches.	UINT	RW	`2`
MWILedTimer	The number of seconds between blinks of the phone port LED to indicate MWI while the phone is onhook. OBi does not blink the phone LED for MWI if this value is set to 0	UINT	RW	`0`

OBIHAI
technology, inc.

PHONE*n* Port – Tip Ring Voltage Polarity (**VoiceService.1.X_FXS.***n***.Polarity.**) for *n* = 1, 2, …, 8				
IdlePolarity	Tip/Ring voltage polarity the line is idle, before a call is connected, or after one side hangs up. Choices are: `Forward`, `Reverse`	Choice	RW	`Forward`
ConnectPolarity	Tip/Ring voltage polarity when the line is connected on a call. Choices are: `Forward`, `Reverse` Note: By using a different polarity for an Idle and a Connected line, OBi effectively generates a polarity reversal signal to the PHONE port, which can be used to signal the attached phone equipment that the call is either connected or ended.	Choice	RW	`Forward`
OriginatingSeizurePolarity	The polarity when making an outgoing call after the line is seized (off-hook) and before the call is connected. Choices are: • `Same As IdlePolarity` • `Same As ConnectedPolarity`	Choice	RW	`Same As IdlePolarity`
TerminatingSeizurePolarity	The polarity when the line is ringing (and the phone is on-hook). Choices are: • `Same As IdlePolarity` • `Same As ConnectedPolarity`	Choice	RW	`Same As IdlePolarity`

Audio Line In

OBi500 has an Audio Line In port which you can connect to an external audio source with a 3.5mm stereo jack. It takes a stereo source and converts it into a mono signal internally and sampled at 8 kHz. This audio source can be used as for MOH. This port is referenced with the short name AI (case-insensitive) in the configuration. To use it for MOH when a phone user places the call on hold, you can specify the phone port parameter **MOHServiceNumber = ai(0)**. AI can answer up to 4 calls simultaneously.

In general, you can route a call to ring ai(0) to let the caller listen to the audio coming from the Audio Line In port.

Line In Port Web Page

Parameter	Description	Type	Access	Default Setting
Line In Port (**VoiceService.1.X_LineIn.1.**)				
MixToLineOut	Enable the mixing of Line In signal to Line Out	Bool	RW	true
InputGain	Line In port input gain in dB (-31 to 24)	INT	RW	0

OBIHAI
technology, inc.

Audio Line Out

OBi500 has an Audio Line Out port which you can connect to an external PA system for paging with a 3.5mm stereo jack. The same audio signal is played on the both left and right channels, sampled at 8 kHz. The audio port is referred to with the short name AO (case insensitive) in the configuration. By default, a phone user can dial #0 to page out of the AO port, with the following rule in the outbound call route: **{#0:ao}**. A similar routing rule may be added elsewhere in the configuration to route a call to send audio out at the AO port. The AO port can only take one call at a time. The OBi also plays a short beep prior to playing caller's voice at the AO port for every call.

Note that Audio Line In source is also looped back to the Audio Out port in the analog domain (and therefore does not subject to quality loss due to ADC/DAC). When paging starts, the OBi automatically lower the level of the analog Audio Line In signal in favor of the caller's audio signal. When paging ends, the OBi gradually raises the level of the analog Audio Line In signal to full volume.

Line Out Port Web Page

Parameter	Description	Type	Access	Default Setting
Line Out Port (**VoiceService.1.X_LineOut.1.**)				
EnablePagingTone	Enable playing Paging Tone at the start of each new page	Bool	RW	false
Volume	Line Out port volume in dB (-47 to 0)	INT	RW	0

Codec Profile Features of the OBi Device

There are two Codec Profiles available on OBi devices. They are selectable Per Trunk (SPn, OBiTALK, $n = 1 - 9$). To select a codec as the preferred codec in this profile, set the priority of that codec to be highest among all the enabled codecs in this profile. Each of the SP and OBiTALK services can be assigned a codec profile in its corresponding configuration. The codec list to use when setting up a call on the underlying service is formed from the list of enabled codecs in the chosen profile and ordered according to the assigned priorities in the profile.

Codec Profile X Web Page (X = A, B)

Parameter	Description	Type	Access	Default Setting
Codec Profile X – G711U Codec (**VoiceService.1.VoiceProfile.1.Line.**n.**Codec.List.1.**) n = 1, 2 corresponding to X = A, B respectively				
Codec	Codec Name	String	RW	G711U
BitRate	Bit rate in bits/sec. Note: Informational only; not configurable	UINT	R	64000
Enable	Enable this codec	Bool	RW	true
SilenceSuppression	Enable silence suppression for this codec	Bool	RW	false
PacketizationPeriod	Packet size in ms	UINT	RW	20
Priority	Priority assigned to this codec (1 is the highest)	UINT	RW	1
PayloadType	Payload type for this codec	UINT	RW	0
FaxPayloadType	The payload type to use for G711U codec when switching over to FAX passthorugh; this can be used as signal to the other end that we have switched to FAX passthrough mode	UINT	RW	0
Codec Profile X – G711A Codec (**VoiceService.1.VoiceProfile.1.Line.**n.**Codec.List.2.**) n = 1, 2 corresponding to X = A, B respectively				
Codec	Codec Name	String	RW	G711A
BitRate	Bit rate in bits/sec Note: Informational only; not configurable	UINT	R	64000
Enable	Enable this codec	Bool	RW	true
SilenceSuppression	Enable silence suppression for this codec	Bool	RW	false
PacketizationPeriod	Packet size in ms	UINT	RW	20
Priority	Priority assigned to this codec (1 is the highest)	UINT	RW	2
PayloadType	Payload type for G711-alaw	UINT	RW	8
FaxPayloadType	The payload type to use for G711A codec when switching over to FAX passthorugh; this can be used as signal to the other end that we have switched to FAX passthrough mode	UINT	RW	8
Codec Profile X – G729 Codec (**VoiceService.1.VoiceProfile.1.Line.**n.**Codec.List.3.**) n = 1, 2 corresponding to X = A, B respectively				
Codec	Codec Name	String	RW	G729
BitRate	Bit rate in bits/sec Note: Informational only; not configurable	UINT	R	8000
Enable	Enable this codec	Bool	RW	true
SilenceSuppression	Enable silence suppression for this codec	Bool	RW	false

PacketizationPeriod	Packet size in ms	UINT	RW	20
Priority	Priority assigned to this codec (1 is the highest)	UINT	RW	3
PayloadType	Payload type for G.729	UINT	RW	18
Codec Profile X – G726R32 Codec (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.List.4.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name	String	RW	G726-32
BitRate	Bit rate in bits/sec Note: Informational only; not configurable	UINT	R	32000
Enable	Enable this codec	Bool	RW	true
SilenceSuppression	Enable silence suppression for this codec	Bool	RW	false
PacketizationPeriod	Packet size in ms	UINT	RW	20
Priority	Priority assigned to this codec (1 is the highest)	UINT	RW	4
PayloadType	Payload type for this codec. Valid range is 96-127	UINT	RW	104
Codec Profile X – iLBC Codec (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.List.8.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name	String	RW	G726-40
BitRate	Bit rate in bits/sec Two values to choose from: 13333 bps or 15200 bps	Choice	RW	13333
Enable	Enable this codec	Bool	RW	true
SilenceSuppression	Enable silence suppression for this codec	Bool	RW	false
PacketizationPeriod	Packet size in ms. Must be multiples of 30 for 13333 bps or multiples of 20 for 15200 bps	UINT	RW	30
Priority	Priority assigned to this codec (1 is the highest)	UINT	RW	5
PayloadType	Payload type for this codec. Valid range is 96-127	UINT	RW	98
Codec Profile X – FAX Event (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.X_FAX.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name for this RTP event, as used in SDP	String	RW	fax-event
Enable	Enable this codec	Bool	RW	false
PayloadType	Payload type to be used to signal RFC2833 FAX tone events. Valid range is 96-127	UINT	RW	100
FaxEvents	A comma separated list of RFC2833 FAX tone events to signal	String	RW	32
Codec Profile X – Telephone Event (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.X_TelephoneEvent.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name for this RTP event, as used in SDP	String	RW	telephone-event
Enable	Enable this codec	Bool	RW	true
PayloadType	Payload type to be used for RFC2833 telephone (DTMF) events. Valid range is 96-127	UINT	RW	101
Codec Profile X – Encap RTP (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.X_EncapRTP.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name. This codec is used to encapsulate RTP packets during a packet loopback call	String	RW	encaprtp

PayloadType	Dynamic Payload type for this codec. Valid range is 96-127	UINT	RW	107
Codec Profile X – Loopback Primer (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.X_LoopbackPrimer.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
Codec	Codec Name. The codec is used by the OBi when acts as a media loopback mirror and before receiving any packets from the loopback source during a media loopback call	String	RW	loopbkprimer
PayloadType	Payload type for this codec. Valid range is 96-127	UINT	RW	108
Codec Profile X – Codec Settings (**VoiceService.1.VoiceProfile.1.Line.***n***.Codec.X_Settings.**) *n* = 1, 2 corresponding to *X* = A, B respectively				
G726BitPacking	Two values to choose from: big-endian or little-endian	Choice	RW	big-endian
T38Enable	Enable the use of T38 (FAX Relay)	Bool	RW	true
T38Redundancy	Reduadancy level for T38 packet transmission. Choices are: 0, 1, 2, 3	Choice	RW	2
T38ECM	Enable T38 ECM	Bool	RW	false
FaxPassThroughCodec	The codec to use for FAX Passthrough. Valid choices are: G711U, G711A	Choice	RW	G711U

Tone Patterns

Note: Tone and Ring Profile A default settings are set for USA telephone standards. Tone and Ring Profile B default settings are set for Australia telephone standards.

Tone Profile Features of the OBi Device

The general format for tone profiles follows the following format: [field-1];[field-2];[field-3];...;[field - 6]

Use ";" to separate the configuration fields.

Note that no spaces are allowed to be used in a tone profile pattern.

Field–1 Composition:

This field describes frequency components used for tone synthesis and it supports up to three different frequencies.

The frequency expression is a string of numeric values with the notation '+' or '-'.

The numeric values are the frequency's decimal values in Hz and amplitude in dBm (Maximum 3 dBm).

Different frequencies are separated by ',' .

Example: 350-18,440-18,550+2

The above example illustrates the 1st frequency at 350 Hz with strength at -18 dBm, the 2nd frequency: 440 Hz with strength at -18 dBm and the 3rd frequency: 550 Hz with strength at +2 dBm.

Field–2 Composition:

This field describes the overall tone playback duration in seconds.

The expression is a numeric value, and supports up to 3 decimated digits.

The numeric value can negative, zero, positive, or skipped:

- Negative value: tone plays indefinitely
- Zero value: tone playback is skipped
- Positive value: Normal playback duration
- No value: tone plays indefinitely

Example: 30.234

  Meaning: tone playback terminates after 30.234 seconds

Field–3 to Field–6 Composition:

Field - 3/4/5/6 share the same definition, and each field describes one single cadence segment. Together 4 fields form a macro-segment, which will be repeated until tone playback expires.

The expression is a string of numeric values with the special notation '/', '(', ')' and ','.

It has a complete format as below:

t(f_0/on_0+off_0,f_1/on_1+off_1,f_2/on_2+off_2,f_3/on_3+off_3)

t: the cadence segment duration in seconds

- Negative value: tone plays indefinitely
- No value: tone plays indefinitely
- Zero value: the duration of this particular segment is zero
- Positive value: Normal playback duration

f_0/1/2/3: a numerical describe which frequency component(s) are used for the synthesis, and it can be one of following 8 options (0 ~ 7)

- 0: No frequency specified, i.e., silent tone
- 1: The 1st frequency
- 2: The 2nd frequency
- 3: The 1st and 2nd frequencies
- 4: The 3rd frequency
- 5: The 1st and 3rd frequencies
- 6: The 2nd and 3rd frequencies
- 7: The 1st and 2nd frequencies if two or more than two frequency components, or the 1st frequency if only one frequency component is available.

If no value is provided for f_0/1/2/3, it will automatically use the combination of the first one or two available frequency components.

on_0/1/2/3: the tone active time in seconds

- Negative value: Not allowed
- No value: infinite tone active time
- Others: normal tone active time (up to 3 decimated digits)

off_0/1/2/3: the tone inactive time in seconds

- Negative value: Not allowed
- No value: infinite tone inactive time
- Others: normal tone inactive time (up to 3 decimated digits)

Example: 4(1/.3+2.34,3/2+1.5)

The above example illustrates using the first frequency to generate tone for 0.3 seconds, followed by 2.34 seconds of silence, then use a combination of the first and second frequencies to generate tone for 2 seconds, then followed by 1.5 seconds silence. The cadence operates repeatedly for 4 seconds.

Tone Examples:

With these examples, we will show the interpretation of a few common tone patterns:

Dial Tone:

DIAL,   "350-18,440-18"

Dial tone is generated as a mixture of two frequency components:

350 Hz at -18 dBm and 440 Hz at -18 dBm

The expiration time is infinite, and tone active time is infinite.

Busy Tone:

BUSY,   "480-18,620-18;10;(.5+.5)"

Busy tone is generated as a mixture of two frequency components:

480 Hz at -18 dBm and 620 Hz at -18 dBm

The expiration time is exactly 10 seconds. It has only one cadence segment, which has tone active 0.5 second and tone inactive 0.5 second.

Prompt Tone:

PROMPT,   "480-16;10"

Prompt tone is generated from a single frequency component:

480 Hz at -16 dBm. The expiration time is exactly 10 seconds. It has only one cadence segment, which has tone infinite active time.

SIT Tone:

SIT_1,   "985-16,1428-16,1777-16;20;(1/.380+0,2/.380+0,4/.380+0,0/0+4)"

Special information tone (SIT) is generated from a set of frequency components:

- 1st frequency: 985 Hz at -16 dBm
- 2nd frequency: 1428 Hz at -16 dBm
- 3rd frequency: 1777 Hz at -16 dBm

The expiration time is exactly 20 seconds. It has only one cadence segment, which includes 4 on-off sections. The segment has infinite repeating time:

- The 1st on-off section: generated by the 1st frequency component, and it has 0.38 tone second active time and 0 inactive time.
- The 2nd on-off section: generated by the 2nd frequency component, and it has 0.38 tone second active time and 0 inactive time.
- The 3rd on-off section: generated by the 3rd frequency component, and it has 0.38 tone second active time and 0 inactive time.
- The 4th on-off section: only generate silence since no frequency component is specified. It has tone 0 second active time and 4 seconds inactive time.

Stutter Tone:

STUTTER,   "350-18,440-18;10;.6(.1+.1);(/)"

Stutter dial tone is generated from a mixture of two frequency components:

350 Hz at -18 dBm and 440 Hz at -18 dBm. The expiration time is exact

10 seconds. It has two cadence segments.

- The first segment: includes only one on-off sections, on 0.1 second and off 0.1 second, and on-off repeats for 0.6 second.
- The second segment: include one on-off section, and has infinite repeating time and infinite tone active time.

Tone Profile X Web Page (X = A, B)

Parameter	Description	Type	Access	Default Setting
Tone Profile X – Dial Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.1.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Dial Tone	String	R	`Dial Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`350-18,440-18;20` (n = 1) `400-18,425-18,450-18;20` (n = 2)
Tone Profile X – Ringback Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.2.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Ringback Tone	String	R	`Ringback Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18,480-18;-1;(2+4)` (n = 1) `400-18,425-18,450-18;` `-1;(.4+.2,.4+2)` (n = 2)
Tone Profile X – Busy Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.3.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Busy Tone	String	R	`Busy Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`480-18,620-18;10;(.5+.5)` (n = 1) `425-18;10;(.4+.4)` (n = 2)
Tone Profile X – Reorder Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.4.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Reorder tone or Fastbusy	String	R	`Reorder or Fastbusy Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`480-18,620-18;10;(.25+.25)` (n = 1) `425-18;10;(.2+.2)` (n = 2)
Tone Profile X – Confirmation Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.5.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Confirmation Tone	String	R	`Confirmation Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`600-18;1;(.2+.2)`
Tone Profile X – Holding Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.6.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Holding Tone played when peer holding the call	String	R	`Holding Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`800-18;30;(.1+10)`
Tone Profile X – Second Dial Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.7.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Second Dial Tone played when dialing second call in a 3-way call	String	R	`Second Dial Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`385-18,484-18;20` (n = 1) `400-18,425-18;20` (n = 2)

Tone Profile X – Stutter Dial Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.8.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Stutter Dial Tone	String	R	`Stutter Tone`

TonePattern	Obihai Tone Pattern Script	String	RW	`350-18,440-18;20;2(.1+.1)`
				`;()               (n = 1)`
				
				`400-18,425-18,450-18;20;`
				`2(.1+.04);()       (n = 2)`

Tone Profile X – Howling Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.9.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Howling Tone for off-hook warning	String	R	`Howling Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`480+3,620+3;10;(.125+.125)`

Tone Profile X – Prompt Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.10.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Prompt Tone to prompt user to enter a number for configuration, such as speed dial	String	R	`Prompt Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`480-16;20`

Tone Profile X – Call Forward DialTone (**VoiceService.1.VoiceProfile.n.Tone.Description.11.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Call Forward Dial Tone (Special dial tone to indicate call-forward-all active)	String	R	`Call Forwarded Dial Tone`
TonePattern	Obihai Tone Pattern Script	String	RW	`350-18,440-18;20;(.2+.2)`

Tone Profile X – Conference Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.12.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Conference Tone (Indicates conference has started)	String	R	`Conference Tone`

TonePattern	Obihai Tone Pattern Script	String	RW	`350-16;10;(.1+.1,.1+9.7)   (n = 1)`
				
				`425-16;10;(1+15,.36+15)   (n = 2)`

Tone Profile X – SIT Tone 1 (**VoiceService.1.VoiceProfile.n.Tone.Description.13.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Special Information Tone - 1	String	R	`SIT Tone 1`

TonePattern	Obihai Tone Pattern Script	String	RW	`985-16,1428-16,1777-16;20;`
				`(1/.380+0,2/.380+0,`
				`4/.380+0,0/0+4)     (n = 1)`
				
				`425-16;20;(2.5+.5)     (n = 2)`

Tone Profile X – SIT Tone 2 (**VoiceService.1.VoiceProfile.n.Tone.Description.14.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Special Information Tone - 2	String	R	`SIT Tone 2`

TonePattern	Obihai Tone Pattern Script	String	RW	`914-16,1371-16,1777-16;20;`
				`(1/.274+0,2/.274+0,`
				`4/.380+0,0/0+4)`

Tone Profile X – SIT Tone 3 (**VoiceService.1.VoiceProfile.n.Tone.Description.15.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Special Information Tone - 3	String	R	`SIT Tone 3`

TonePattern	Obihai Tone Pattern Script	String	RW	`914-16,1371-16,1777-16;20;`
				`(1/.380+0,2/.380+0,`
				`4/.380+0,0/0+4)`

Tone Profile X – SIT Tone 4 (**VoiceService.1.VoiceProfile.n.Tone.Description.16.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Special Information Tone - 4	String	R	`SIT Tone 4`

TonePattern	Obihai Tone Pattern Script	String	RW	`985-16,1371-16,1777-16;20;`
				`(1/.380+0,2/.380+0,`
				`4/.380+0,0/0+4)`

Tone Profile X – Outside Dial Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.17.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Outside Dial Tone	String	R	Outside Dial Tone
TonePattern	Obihai Tone Pattern Script	String	RW	385-16;10

Tone Profile X – R-Command Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.18.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	R-Command Tone (The tone played to prompt user to enter a command digit 0-4 after pressing the 'R' key, when using Nordic style mid-call commands)	String	R	R-Command Tone
TonePattern	Obihai Tone Pattern Script	String	RW	400-16;5

Tone Profile X – Paging Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.19.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	Paging Tone (The short tone played before playing the caller's voice)	String	R	Paging Tone
TonePattern	Obihai Tone Pattern Script	String	RW	480-16;1;(.2+.2)

Tone Profile X – DND Dial Tone (**VoiceService.1.VoiceProfile.n.Tone.Description.20.**) for n = 1, 2 corresponding to X = A, B respectively				
ToneName	DND Dial Tone (Special dial tone to indicate DND active)	String	R	DND Dial Tone
TonePattern	Obihai Tone Pattern Script	String	RW	350-18,440-18;20;(.2+.2)

Ring Pattern

The general format of an OBi Ring Profile is as follows: [field-1];[field-2];...;[field - 5]

Use the ";" to separate up to five (5) configuration fields.

Please note that no spaces are allowed to be used in a tone profile pattern.

Field–1 Composition:

Field-1 describes the overall ringing duration in seconds.

The expression is a numeric value, and supports up to 3 decimated digits.

The numeric value can negative, zero, and positive:

- Negative value: Ringing lasts indefinitely
- No value: Ringing lasts infinitely
- Zero value: Ringing is skipped
- Positive value: Normal ringing duration

Example: 30.5

The above example illustrates a ringing tone that terminates after 30.5 seconds.

Field –2 to Field –5 Composition:

Field - 2/3/4/5 share the same definition, and each field describes one single cadence segment. Together, the four (4) fields form a macro-segment, which will be repeated until ringing expires.

The expression is a string of numeric values with the special notation '**(**' , '**)**' and '**,**'

It has the format as per the following construct: $t(on\_0+off\_0,on\_1+off\_1,on\_2+off\_2,on\_3+off\_3)$

t: The cadence segment duration in seconds.

- Negative value: Ringing indefinitely
- No value: Ringing indefinitely
- Zero value: Ringing is skipped
- Positive value: Normal ringing duration

on_0/1/2/3: The ring active time in seconds.

- Negative value: Not allowed
- No value: Infinite ring active time
- Others: Normal ring active time (up to 3 decimated digits)

off_0/1/2/3: The ring inactive time in seconds

- Negative value: Not allowed
- No value: Infinite ring inactive time
- Others: Normal ring inactive time (up to 3 decimated digits)

Example: 4(.3+2.34,2+1.5)

The above example illustrates a ringing tone comprised of two segments. Ringing is active for 0.3 seconds, followed by 2.34 seconds of silence, then ringing for 2 seconds, and followed by 1.5 seconds of silence.

The above cadence operates repeatedly for 4 seconds.

Ring Profile X Web Page (X = A, B)

Parameter	Description	Type	Access	Default Setting
Ring Profile X – Call Waiting Tone 1 (**VoiceService.1.VoiceProfile.1.Tone.Description.21.**)				
ToneName	Distinctive Call Waiting Tone 1	String	RW	`Bellcore-dr1`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.25+10)` (n = 1) `425-18;30;(.2+.2,.2+4.4)` (n = 2)
Ring Profile X – Call Waiting Tone 2 (**VoiceService.1.VoiceProfile.1.Tone.Description.22.**)				
ToneName	Distinctive Call Waiting Tone 2	String	RW	`Bellcore-dr2`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.1+.1,.3+.1,.1+10)`
Ring Profile X – Call Waiting Tone 3 (**VoiceService.1.VoiceProfile.1.Tone.Description.23.**)				
ToneName	Distinctive Call Waiting Tone 3	String	RW	`Bellcore-dr3`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.1+.1,.1+10)`
Ring Profile X – Call Waiting Tone 4 (VoiceService.1.VoiceProfile.1.Tone.Description.24.)				
ToneName	Distinctive Call Waiting Tone 4	String	RW	`Bellcore-dr4`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.1+.1,.1+.1,.1+10)`
Ring Profile X – Call Waiting Tone 5 (**VoiceService.1.VoiceProfile.1.Tone.Description.25.**)				
ToneName	Distinctive Call Waiting Tone 5	String	RW	`Bellcore-dr5`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.3+.1,.1+.1,.3+10)`
Ring Profile X – Call Waiting Tone 6 (**VoiceService.1.VoiceProfile.1.Tone.Description.26.**)				
ToneName	Distinctive Call Waiting Tone 6	String	RW	`User-dr1`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.1+.1,.3+.2,.3+10)`
Ring Profile X – Call Waiting Tone 7 (**VoiceService.1.VoiceProfile.1.Tone.Description.27.**)				
ToneName	Distinctive Call Waiting Tone 7	String	RW	`User-dr2`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.3+.1,.3+.1,.1+10)`
Ring Profile X – Call Waiting Tone 8 (**VoiceService.1.VoiceProfile.1.Tone.Description.28.**)				
ToneName	Distinctive Call Waiting Tone 8	String	RW	`User-dr3`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.3+2)`
Ring Profile X – Call Waiting Tone 9 (**VoiceService.1.VoiceProfile.1.Tone.Description.29.**)				
ToneName	Distinctive Call Waiting Tone9	String	RW	`User-dr4`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.3+2)`
Ring Profile X – Call Waiting Tone 10 (**VoiceService.1.VoiceProfile.1.Tone.Description.30.**)				
ToneName	Distinctive Call Waiting Tone 10	String	RW	`User-dr5`
TonePattern	Obihai Tone Pattern Script	String	RW	`440-18;30;(.3+2)`
Ring Profile X – Ring Pattern 1 (**VoiceService.1.VoiceProfile.1.Line.n.Ringer.Description.1.**) for n = 1, 2 corresponding				

to X = A, B respectively					
RingName		String	RW	Bellcore-dr1	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(2+4) (*n* = 1)

60;(.4+.2,.4+2)   (*n* = 2) | |
Ring Profile X – Ring Pattern 2 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.2.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	Bellcore-dr2	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.3+.2,1+.2,.3+4)	
Ring Profile X – Ring Pattern 3 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.3.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	Bellcore-dr3	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.8+.4,.8+4)	
Ring Profile X – Ring Pattern 4 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.4.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	Bellcore-dr4	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.4+.2,.3+.2,.8+4)	
Ring Profile X – Ring Pattern 5 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.5.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	Bellcore-dr5	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.2+.2,.2+.2,.2+.2,1+4)	
Ring Profile X – Ring Pattern 6 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.6.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	User-dr1	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.2+.4,.2+.4,.2+4)	
Ring Profile X – Ring Pattern 7 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.7.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	User-dr2	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.4+.2,.4+.2,.4+4)	
Ring Profile X – Ring Pattern 8 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.8.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	User-dr3	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.25+9.75)	
Ring Profile X – Ring Pattern 9 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.9.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	User-dr4	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.25+9.75)	
Ring Profile X – Ring Pattern 10 (**VoiceService.1.VoiceProfile.1.Line.***n***.Ringer.Description.10.**) for *n* = 1, 2 corresponding to X = A, B respectively					
RingName		String	RW	User-dr5	
RingPattern	OBIHAI Ring Pattern Script	String	RW	60;(.25+9.75)	

Star Code Profile Features of the OBi Device

Star codes are short sequences of digits where each sequence serves as a command to the OBi Device to perform certain operation. Each sequence usually starts with the * key followed by a 2-digit code (such as *69), hence the term star code. A typical operation to carry out is to set the value of one or more configuration parameters. At present the OBi device allows user to issue star code from the PHONE port only; user issues a star code the same way he dials a number to make a call. In OBi every star code and its operation are defined with a short *Star Code Script* parameter. The set of star codes that can be dialed from the PHONE port is collectively referred to as a Star Code Profile.

OBi5xx has nine (9) star code profiles available in its configuration, known as Start Code Profile A, B, …, H respectively. Each profile has 30 star code script parameters, known as Code1 to Code30. You can select which star code profile to use by setting **PHONE Port::**`StarCodeProfile` to *A, B, …* or *H*, or *None* if star code is not to be used. Hence each phone port may use a different Star Code Profile.

A star code script is defined with the help of a number of predefined variables and actions. Each variable represents one or one group of configuration parameters. An action can be checking or setting the value of a variable, collecting a phone number from the user, or calling a certain number.

Star Code Script Variables (*VAR*)

A star code script variable or *VAR* can be trunk specific or global (non-trunk specific). The general format of a global variable is $var. The general format of a trunk specific variable is *TK*($var) , where *TK* is the abbreviated name of a trunk (SP1, SP2, LI1, or PP1). If *TK* is not specified for a trunk-specific variable, it implies all the applicable trunks in the system.

Note that: SP1 is the SP1 Service, SP2 the SP2 Service, LI1 the LINE port, and PP1 the OBiTALK Service. Each service is also referred to as a "trunk" in this document.

Here is a list of the supported $var:

$CFA = call forward unconditional enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$CFB = call forward busy enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$CFN = call forward no-answer enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$CFAN = call forward unconditional number (trunk specific; admissible value: a token representing a call forward number)

$CFBN = call forward busy number (trunk specific; admissible value: a token representing a call forward number)

$CFNN = call forward no-answer number (trunk specific; admissible value: a token representing a call forward number)

$MWS = message waiting state (trunk specific; admissible value: 0 for no new messages, 1 for one or more new messages)

$DND = do-not-disturb enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$BAC = block-anonymous caller enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$BCI = block outbound caller-ID enable (trunk specific; admissible value: 0 for disable, 1 for enable)

$CWA = call-waiting enable (global; admissible value: 0 for disable, 1 for enable)

$BCI1 = block caller-ID once (global; admissible value: 1 for enable)

$UBCI1 = unblock caller-ID once (global; admissible value: 1 for enable)

$LBM1 = Loopback media (audio samples) once in the next call

$LBP1 = Loopback RTP packets once in the next call

$CDM1 = Codecs to enable in the next call (temporarily overriding any codec preferences in device configuration). Each bit of its value represents one audio codec:

- Bit0 (LSB) = G711u

- Bit1 = G711a

- Bit2 = G726r16

- Bit3 = G726r24

- Bit4 = G726r32

- Bit5 = G726r40

- Bit6 = G729

$LDN = last dialed number (for redial) (global; read only)

$BAR1 = Enable Barge-In 1 on the next call (global; admissible value: 1 for enable)

$Bxrn = Blind Transfer Target Number (global; admissible value: a token representing the target number)

$LCR = last caller's number (for call return) (global; read only)

$SPD[*n*] = number for the speed dial *n* (*n* = 1 – 99) (global; admissible value: literal or token representing a phone number)

$CODE = the digit(s) representing the variable part of a star code (see examples below; read only)

Variable names are CASE INSENSITIVE.

Star Code Script Actions (*ACT*)

The general format of an action: *ACT*(*par, par*, ...)

The following actions are supported:

- set(*VAR,token*) = Set the given *VAR* to the value represented by *token*.
- call(*token*) = Call the number represented by *token.*
    - **PHONE Port::**OutboundCallRoute will be applied when making the call (but not the DigitMap)
- rpdi(*token*) = repeat dial the number represented by *token*
- coll(*VAR*) = collect a number from the user and store it as the value of the parameter(s) represented by *VAR*.
    - The number is collected with **PHONE Port::**DigitMap applied
- say(*token*) = announce the value represented by *token*
    - Values are announced as a list of alphabets or numbers
    where *token* can be a literal (such as 1234) or another variable (such as $CFAN or SP1($CFBN))
- btdscvr(*n*) = Enable OBiBT discovery for 2 minutes
    - *n* = 0 or 1 for OBiBLueTooth 1 and OBiBlueTooth 2 respectively

Action names are CASE INSENSITIVE.

Star Code Script Format

General Format: code, name, action1, action2, action3, …

- code = the star code, such as *72. It may contain a variable part enclosed in parenthesis, such as *74(x|xx)
    - The variable part as entered by the user are stored in the variable $CODE
- name = a descriptive name of the function of this star code, such as *Call Forward Unconditional*
- action1, action2, … = a valid action with parameters

Actions are carried out one-by-one in the order as specified in the script.

Restrictions:

- At most 1 *coll* action per code.
- Either 1 *say* or 1 *call* action at most per code, and it must be the last action in the script.

Star Code Script Examples

The following examples are taken from some of the default star code scripts in the OBi device.

*69, Call Return, call($LCR)

- Calls the number of the caller who rings the PHONE port last time

*07, Redial, call($Ldn)

- Redials the last dialed number

*72, Call Forward Unconditional, coll($cfan),set($cfa,1)

- Collects a number from the user according to the DigitMap. Then set the CallForwardUnconditionalNumber on all trunks to the collected value, and set the CallForwardUnconditionalEnable on all trunks to Yes
- To modify the script to enable CallForwardUnconditional on SP1 only, change it to

*72, Call Forward Unconditional SP1, coll(SP1($cfan)),set(SP1($cfa),1)

*67, Block Caller ID Once, set($BCI1,1)

- Enable masking of caller ID information once for the next call on any trunk

*74(x|xx), Set Speed Dial, coll($Spd[$code])

- After user dials *74, OBi expects one or two more digits from the user which represent a speed dial slot index (1 to 99). The 1 or 2-digit variable part is stored in the variable $code.
- OBi device then plays a prompt tone and proceeds to collect a number from the user according to the DigitMap. Finally OBi stores the collected number in the given speed dial slot. If the slot already has a number specified, it will be overwritten quietly with the new value.

*75(x|xx), Check Speed Dial, say($Spd[$code])

- After user dials *75, OBi expects one or two more digits from the user which represent a speed dial slot index (1 to 99). The 1 or 2-digit variable part is stored in the variable $code.
- OBi device then announces the number stores in the speed dial slot, or says "not available" if the slot is empty.

Star Code Profile X Web Page (X = A, B, C, D, E, F, G, H)

Parameter	Description	Type	Access	Default Setting
Star Code Profile *X* **– Star Codes** (**VoiceService.1.X_StarCode.***n*.) for *n* = 1, 2,..8 corresponding to *X* = A, B, …, H				
Code1	Default = Redial Star Code	String	RW	`*07, Redial, call($Ldn)`
Code2	Default = Call Return Star Code	String	RW	`*69, Call Return, call($Lcn)`
Code3	Default = Block Caller ID (Persistent) Star Code	String	RW	`*81, Block Caller ID, set($Bci,1)`
Code4	Default = Unblock Caller ID (Persistent) Star Code	String	RW	`*82, Unblock Caller ID, set($Bci,0)`
Code5	Default = Block Caller ID Once Star Code	String	RW	`*67, Block Caller ID Once, set($Bci1,1)`
Code6	Default = Unblock Caller ID Once Star Code	String	RW	`*68, Unblock Caller ID Once, set($Ubci1,1)`
Code7	Default = Call Forward Unconditional Star Code	String	RW	`*72, Cfwd All, coll($Cfan), set($Cfa,1)`
Code8	Default = Disable Call Forward Unconditional Star Code	String	RW	`*73, Disable Cfwd All, set($Cfa, 0)`
Code9	Default = Call Forward on Busy Star Code	String	RW	`*60, Cfwd Busy, coll($Cfbn), set($Cfb,1)`
Code10	Default = Disable Call Forward on Busy Star Code	String	RW	`*61, Disable Cfwd Busy, set($Cfb, 0)`
Code11	Default = Call Forward on No Answer Star Code	String	RW	`*62, Cfwd No Ans, coll($Cfnn), set($Cfn,1)`
Code12	Default = Disable Call Forward on No Answer Star Code	String	RW	`*63, Disable Cfwd No Ans, set($Cfn,0)`
Code13	Default = Block Anonymous Calls Star Code	String	RW	`*77, Block Anonymous Call, set($Bac,1)`
Code14	Default = Unblock Anonymous Calls Star Code	String	RW	`*87, Unblock Anonymous Call, set($Bac,0)`
Code15	Default = Enable Call Waiting Star Code	String	RW	`*56, Enable Call Waiting, set($Cwa,1)`
Code16	Default = Disable Call Waiting Star Code	String	RW	`*57, Disable Call Waiting, set($Cwa,0)`
Code17	Default = Do Not Disturb Star Code	String	RW	`*78, Do Not Disturb, set($Dnd,1)`
Code18	Default = Disable Do Not Disturb Star Code	String	RW	`*79, Disable DND, set($Dnd,0)`
Code19	Default = Repeat Dial Star Code	String	RW	`*66, Repeat Dial, rpdi($Ldn)`
Code20	Default = Disable Repeat Dial Star Code	String	RW	`*86, Disable Repeat Dial, rpdi`
Code21	Default = Set Speed Dial Star Code	String	RW	`*74(x|xx), Set Speed Dial, coll($Spd[$Code])`
Code22	Default = Check Speed Dial Star Code	String	RW	`*75(x|xx), Check Speed Dial, say($Spd[$Code])`
Code23	Default = Loopback Media Star Code	String	RW	`*03, Loopback Media, set($Lbm1,1)`
Code24	Default = Loopback RTP Star Code	String	RW	`*04, Loopback RTP Packet, set($Lbp1,1)`
Code25	Default = Force G711u Codec Star Code	String	RW	`*4711, Use G711 Only, set($Cdm1,3)`
Code26	Default = Force G729 Codec Star Code	String	RW	`*4729, Use G729 Only, set($Cdm1,4)`
Code27	Default = Clear Speed Dial Star Code	String	RW	`*76([1-9]|[1-9]x), Clear Speed Dial, set($Spd[$Code],)`
Code28	Default = Blind Transfer Star Code	String	RW	`*98, Blind Transfer, coll($Bxrn)`
Code29	Default = Barge In Star Code	String	RW	`*96, Barge In, set($Bar1,1)`
Code30	Default = Make OBiBlueTooth 1 discoverable (for 2 minutes)	String	RW	`*28, OBiBT Discoverable, btdscvr(0)`
Code31		String	RW	
Code32	Default = Set OBiPLUS system to day mode	String	RW	`*10, Day Mode, set($Opm,0)`
Code33	Default = Set OBiPLUS system to night mode	String	RW	`*11, Night Mode, set($Opm,1)`
Code34	Default = Set OBiPLUS system to auto	String	RW	`*12, Auto Night Mode, set($Opm,2)`

OBIHAI
technology, inc.

	day/night mode based on office hour settings			
Code35	Default = Make OBiBlueTooth 2 discoverable (for 2 minutes)	String	RW	*29, OBiBT Discoverable, btdscvr(1)
Code36		String	RW	
Code37		String	RW	
Code38		String	RW	
Code39		String	RW	
Code40		String	RW	

User Settings Features of the OBi Device

Speed Dial Numbers

Each OBi device supports 99 speed dial numbers. The 99 speed dial slots are numbered from 1 to 99 and are invoked by dialing a 1 or 2-digit number corresponding to the slot number. Speed dials may be dialed from the PHONE port or via the Auto Attendant. Note that the 2-digit numbers "01", "02", …, "09" are not admissible; you must dial the 1-digit number "1", "2", …, "9" for slot number 1-9.

Speed dial value can be set using the configuration web page, remote provisioning, or star code (see the *Star Code Section* in this document for more details). The value may be a number just like the one you normally dial, with or without any service access code prefix, such as: **9200112233, **214089991123, 4280913, etc. It may also include explicit trunk information with the general format TK(number), where TK= SPn (n=1 – 8), BT1, BT2, or PP. For example, PP(ob200112233), SP2(14089991123), BT2(4280913), etc.

If trunk information is *not* specified in the speed dial entry, OBi device applies DigitMap and OutboundCallRoute when making the call. Otherwise neither DigitMap nor OutboundCallRoute is applied.

Using Speed Dial Number as Ad Hoc Gateway

If an external gateway does not require authentication, its access number can be stored in one of the 99 speed dial slots to allow ad hoc direct dialled gateway calls. To do this, the user dials the gateway's speed dial, followed by a *, followed by the target number. That is <gateway-speeddial> * <target-number>. For example, the gateway access number pp(ob200333456) is stored at speed dial 8, and the user can dial 8*14085551234 to call 14085551234 using the given gateway.

Note: At the present time, only gateways that are accessed with an OBi number can be used this way.

Speed Dials Web Page

Parameter	Description	Type	Access	Default Setting
Speed Dials (**SpeedDial.**)				
N (Repeat for N = 1 – 99)	Speed Dial N	String	RW	

User Defined Digit Maps Web Page

For an explanation of this feature, please refer to the section User Defined Digit Maps under Digit Map Configuration.

Parameter	Description	Type	Access	Default Setting
User Defined Digit Map 1 (**UserDigitMap.1.**)				
Label	A 2-16 characters long label to reference this digit map in other digit maps and call routing rules. It must be alphanumeric, not containing any spaces, and different from other user-defined or built-in digit map labels	String	RW	`ipd`

DigitMap	A valid digit map. Default Setting:
	
	 `(xx.<*:@>xx?x?<*:.>xx?x?<*:.>xx?x?<*:.>xx?x?`
	 `|xx.<*:@>xx?x?<*:.>xx?x?<*:.>xx?x?<*:.>xx?x?`
	 `<*::>xx?x?x?x?)`
	
	 Note: this default value supports IPv4 dialing
		String	RW	See Desc.

User Defined Digit Map *n* (**UserDigitMap.***n***.**) for *n* = 2 – 10				
Label	A 2-16 characters long label to reference this digit map in other digit maps and call routing rules. It must be alphanumeric, not containing any spaces, and different from other user-defined or built-in digit map labels	String	RW	
DigitMap	A valid digit map	String	RW	

OBIHAI
technology, inc.

OBi Call Routing and Digit Map

Trunks, Endpoints, and Terminals

An OBi device is a Voice Service Bridge (VSB) that supports multiple voice services. It can bridge calls across any of the supported services. By a call bridge we refer to a voice connection connecting two calls on the same or different voice services. The OBi500 allows 8 concurrent independent call bridges. The following matrix shows the possible call bridge connections.

Supported 2-way call bridges on the OBi500

	SP1 Service	SP2 Service	SP3 Service	SP4 Service	SP5 Service	SP6 Service	SP7 Service	SP8 Service	OBiTALK Service	OBiBlue Tooth 1	OBiBlue Tooth 2
SP1 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP2 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP3 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP4 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP5 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP6 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP7 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
SP8 Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
OBiTALK Service	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
OBiBlueTooth 1	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes
OBiBlueTooth 2	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	no

Each supported service is also referred to as a *trunk* (a traditional telco term for a physical wire or wires that deliver phone services to homes or businesses). Each trunk is represented with 2-letter abbreviation and a 1-based instance identifier:

- SP1 = the SP1 Voice Service (with ITSP A, B, C, or D)
- SP2 = the SP2 Voice Service (with ITSP A, B, C, or D)
- SP3 = the SP3 Voice Service (with ITSP A, B, C, or D)
- SP4 = the SP4 Voice Service (with ITSP A, B, C, or D)
- SP5 = the SP5 Voice Service (with ITSP A, B, C, or D)
- SP6 = the SP6 Voice Service (with ITSP A, B, C, or D)
- SP7 = the SP7 Voice Service (with ITSP A, B, C, or D)
- SP8 = the SP8 Voice Service (with ITSP A, B, C, or D)
- PP1 = the OBiTALK Service
- BT1 = the OBiBlueTooth 1 Service (with OBiBT dongle connected to USB Port 1)
- BT2 = the OBiBlueTooth 2 Service (with OBiBT dongle connected to USB Port 2)

The instance identifier may be omitted if it is equal to 1; hence BT is equivalent BT1, PP is equivalent to PP1, etc. These short-hand notations are used heavily in configuring the OBi device, as found in call routes, call forward numbers, and speed dials parameters. Unless stated otherwise, the abbreviated trunk names are case insensitive.

The PHONE ports, AA, AO, AI, and PK are the entities in an OBi500 device that calls can terminate (i.e., starts or ends there), as opposed to the trunks, which rely on the corresponding service providers to terminate the call. In this document we refer to the PHONE ports, AA, AO, AI, and PK as *endpoints*. Like the trunks, each endpoint is represented by a 2-letter abbreviation and a 1-based instance identifier:

-   PH*n* = The PHONE *n* Port (*n* = 1, 2, …, 8)
-   AA = The Auto Attendant
-   AI = The Audio (Line) In Port
-   AO = The Audio (Line) Out Port
-   PK = The Call Park

Unless stated otherwise, abbreviated endpoint names are case insensitive. A trunk or an endpoint is also referred to as a *Terminal* in this document.

The following matrix shows the possible call connections between the endpoints and the trunks:

Supported endpoint calls on the OBi

	Any Trunk	PH1	PH2	PH3	PH4	PH5	PH6	PH7	PH8	AA	AI (callee)	AO (callee)	PK (callee)
Any Trunk	n/a	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
PH1	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
PH2	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes	yes
PH3	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes	yes
PH4	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	yes	yes
PH5	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes	yes
PH6	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes	yes
PH7	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes	yes
PH8	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes	yes
AA	yes	yes	yes	yes	yes	yes	yes	yes	yes	no	yes	yes	yes
AI (caller)	no	no	no	no	no	no	no	no	no	no	no	no	no
AO (caller)	no	no	no	no	no	no	no	no	no	no	no	no	no
PK (caller)	no	no	no	no	no	no	no	no	no	no	no	no	no

Call Routing – The OBi Way

Call Routing is the process by which the OBi Device sets up a call bridge or a (endpoint) call based on such information as: the trunk on which the call originates, the caller's number, the called number, etc. Call Routing Rules are parameters used to instruct the OBi device how to route calls. A call may transform into a call bridge or an endpoint call after being routed by the OBi according to the given routing rules.

Every call has to be originated from somewhere. From the device's perspective, calls originated from the trunk side are considered Inbound Calls, while calls originated from an endpoint Outbound Calls. The call routing rule syntaxes for inbound calls and outbound calls are slightly different and we shall explain them separately below. Call Routing Rule configuration relies heaving on digit maps. If you are not familiar with how digit map works yet, please read the *Digit Map Configuration Section* in this document first.

Inbound Call Route Configuration

Every trunk has a corresponding InboundCallRoute in the OBi device configuration. It is a comma separated list of rules where each rule is also surrounded by a pair of curly braces { }. No extra white spaces are allowed. These rules tell the OBi how to handle an inbound call, such as sending it to the PHONE port (and ringing the attached phone(s)), sending it to the Auto Attendant for further routing (interactively with the caller), or making another call on a specific trunk to bridge with this call.

The general format is:

InboundCallRoute := *rule* **OR** {*rule*},{*rule*},….

Note that the curly braces may be omitted if there is only one rule in the route. The **OR** operator is NOT part of the parameter syntax; it is used here to separate alternative values only.

A rule has the following format:

rule := peering-list : terminal-list

where

peering-list := *peering,peering,…*	(comma separated list of 0 or more *peering* object)
terminal-list := *terminal,terminal,….*	(comma separated list of 0 or more *terminal* object)

peering := caller-list > callee-list

caller-list := *caller|caller|caller|…*	(vertical bar separated list of 0 or more *caller* object)
callee-list := *callee|callee|callee|…*	(vertical bar separated list of 0 or more callee object)
caller := *number* **OR** _embedded-digit-map_ **OR** ? **OR** @	(?=anonymous, @=any number but anonymous)

callee := number **OR** embedded-digit-map **OR** @

terminal := PH*x* **OR** AA*x* **OR** LI*x*(*arg*) **OR** SP*x*(*arg*) **OR** PP*x*(*arg*)	(*arg* object is optional)

arg := cid > target

x := 1 OR 2 OR 3…	(where applicable; can be omitted if it is equal to 1)

cid := spoofed-caller-number **OR** $1

target := number-to-call **OR** $2

embedded-digit-map := (Mlabel) **OR** digit-map

Notes:

- *Terminal-list* can be empty, which means to block this call. The preceding ':' cannot be omitted. Up to 4 *terminals* may be specified in the list. The listed *terminals* will be called/rung by OBi simultaneously; we refer to this operation as *forking* the call. A terminal may be a trunk or an endpoint.
- Abbreviated terminal names are case-insensitive

- *number and number-to-call* are literal strings, such as 14089991234
- *digit-map* is just any proper digit map, such as (1xxx|xx.); make sure to include the enclosing parentheses
- *spoofed-caller-number* is a literal string, such as 14081112233, to be used as the caller number for making a new call on the specified trunk
- (M*label*) is a named digit map, where *label* is the abbreviated name of any terminal that has a digit map defined: SP1–SP8, BT1, BT2, PP, PH1–PH8
- $1 is an internal variable containing the value of the caller number of this inbound call, after any digit map transformation in the matched *caller* object of the matched *peering* object in the *peering-list*.
- $2 is an internal variable containing the called number of this inbound call, after any digit map transformation in the matched *callee* object of the matched *peering* object in the *peering-list*.

More notes on *peering-list* and *peering* objects:

- *Peering-list* is optional in InboundCallRoute. If *peering-list* is empty, the succeeding ':' can be omitted also. An empty *peering-list* implies a single *peering* object whose *caller* object list matches any caller number. That is, the InboundCallRoutes listed below are all equivalent
    - o   ph
    - o   {ph}
    - o   {:ph}
    - o   {?|@>@:ph}
- *Callee-list* in a *peering* object can be empty. It implies the *callee* object @, meaning any called number. The preceding '>' can be omitted if callee-list is empty.
- *Caller-list* in a *peering* object can be empty. It implies the *caller-list* @|?, meaning any caller number including anonymous. The succeeding '>' cannot be omitted if caller-list is empty but not the callee-list

More notes on the *arg*, *cid*, and *target* objects:

- The *cid* object inside an *arg* object is optional. If omitted, it implies no caller-ID spoofing when making the call on the specified trunk. The succeeding '>' can be omitted is *cid* is omitted
- The *target* object inside an *arg* object is optional. If omitted, it implies the *target* $2, which means to call the original called number after applying any necessary digit map transformation implied by the rule. The preceding '>' cannot be omitted if *target* is omitted but cid is not
- *arg* object is optional. If omitted, it implies the *arg* with the *target* $2 and no *cid*. If *arg* is omitted, the succeeding parentheses ( ) can be omitted also.

An inbound call matches a rule if its caller-number/callee-number matches one of the *peering* objects of the rule. *Peering* objects are tested in the order left and right, and the first matched *peering* object will win. Rules are also checked in the order left to right, and the first matched rule will win. Therefore it is important that you place the more specific rules first in the InboundCallRoute if multiple rules can potentially match the same inbound call.

InboundCallRoute Examples:

1)   ph **OR** {ph} **OR** {:ph} **OR** {@|?>@:ph}   (all equivalent)

It says: Ring the PHONE port for all incoming calls. This is the default InboundCallRoute for all trunks.

2)   {14081223330|15103313456:ph,aa},{(1800xx.|1888xx.)|?:},{ph}

OBIHAI
technology, inc.

It says: Ring both PHONE port and AA for calls coming from 1 408 122 3330 or 1 510 331 3456, block all 800, 888, and anonymous calls, and ring the PHONE port for all other calls

3) {(x.4081113333|x.4152224444):aa},{ph}

It says: Ring the AA for calls coming from any number that ends with 408 111 3333 or 415 222 4444, and ring the PHONE port for all other calls. Be sure to include the enclosing parentheses in this example since "x." is a digit map specific syntax.

4) {200123456:aa},{sp1(14083335678)}

It says: Ring the AA for calls coming from 200123456. For all any other call, bridge it by calling 1 408 333 5678 using SP1 Service

Outbound Call Route Configuration

Every endpoint has an OutboundCallRoute parameter in the OBi device configuration. It tells the device where to send the call when the endpoint attempts to make a call. Endpoints may call each other or an outside number using one of the trunks. The OutboundCallRoute syntaxes are almost identical to those of the InboundCallRoute; the differences are mainly in the implied value when an optional field is omitted, no *caller* objects and one and only one terminal object per terminal-list in an OutboundCallRoute. Forking is not supported when routing outbound calls.

The general format is:

OutboundCallRoute := *rule* **OR** {*rule*},{*rule*},….

Note that the curly braces may be omitted if there is only one rule in the route. The **OR** operator is NOT part of the parameter syntax; it is used here to separate alternative values only.

A rule has the following format:

rule := callee-list : terminal

where

callee-list := *callee*|*callee*|*callee*| …   (vertical bar separated list of 0 or more callee object)

callee := *number* **OR** *embedded-digit-map* **OR** @   (@ = any number)

terminal := PH*x* **OR** AA*x* **OR** LI*x*(*arg*) **OR** SP*x*(*arg*) **OR** PP*x*(*arg*)   (*arg* object is optional)

arg := cid > target

    x := 1 OR 2 OR 3…   (where applicable; can be omitted if it is equal to 1)

cid = spoofed-caller-number

target = number-to-call **OR** $2

embedded-digit-map = (Mlabel) **OR** digit-map

Notes:

- A terminal may be a trunk or another endpoint.
- Abbreviated terminal names are case-insensitive
- *number and number-to-call* are literal strings, such as 14089991234
- *digit-map* is just any proper digit map, such as (1xxx|xx.); make sure to include the enclosing parentheses
- *spoofed-caller-number* is a literal string, such as 14081112233, to be used as the caller number for making a new call on the specified trunk
- (M*label*) is a named digit map where *label* is the abbreviated name of any terminal that has a digit map defined: SP1–SP8, BT1, BT2, PP, PH1-PH8, or AA
- $2 is an internal variable containing the called number of this outbound call, after any digit map transformation in the matched *callee* object
- *Callee-list* can be empty, which implies the single *callee* object @, which means any called number. The succeeding ':' can be omitted also when *callee-list* is empty

More notes on the *arg*, *cid*, and *target* objects:

- The *cid* object inside an *arg* object is optional. If omitted, it implies no caller-ID spoofing when making the call on the specified trunk. The succeeding '>' can be omitted if *cid* is omitted.
- The *target* object inside an *arg* object is optional. If omitted, it implies the *target* $2, which means to call the original called number after applying any necessary digit map transformation implied by the rule. The preceding '>' cannot be omitted if *target* is omitted but not the cid.
- *arg* object is optional. If omitted, it implies the *arg* with the *target* $2 and no *cid*

An outbound call matches a rule if it's called number matches one of the *callee* objects of the rule. *Callee* objects are tested in the order left and right, and the first matched *callee* will win. Rules are also checked in the order left to right, and the first matched rule will win. Therefore it is important that you place the more specific rules first in the OutboundCallRoute if multiple rules can potentially match the same outbound call.

Note that every endpoint also has a digit map defined. The user dialed number is completely processed with the endpoint's digit map first before it is passed to the OutboundCallRoute for routing decision. Therefore the number used for matching call routing rules has already incurred the transformations, if any, implied by the digit map. Remember this fact when crafting your own OutboundCallRoute.

OutboundCallRoute Examples:

1) sp1 **OR** {SP1} **OR** {:SP1} **OR** {@:Sp1}   (all equivalent)
This rule says: Make all calls using SP1 Service, without any caller-id spoofing or digit transformation

2)
{(<#:>|911):li},{**0:aa},{***:aa2},{(Mpli):pli},{(<**1:>(Msp1)):sp1},{(<**2:>(Msp2)):sp2},{(<**8:>(Mli)):li},{(<**9:>(Mpp)):pp}
This is the default OutboundCallRoute for the PHONE port. It says:

OBIHAI
technology, inc.

- Dial # key to connect to the LINE port; OBi will take the LINE port hardware to off-hook but will not dial out any digit. The net result is that the user will hear dial tone generated by the PSTN company if a working line is connected to the LINE port. The user may then continue to dial the PSTN number directly to the phone company. Note that from the OBi's point of view, the PSTN call is connected the moment it takes the LINE port to the off-hook state!
- Use the PSTN Line to call 911
- Dial **0 to invoke AA1 or AA
- Dial *** to invoke the local device configuration IVR (a.k.a. AA2)
- (Mpli) and pli will be substituted with the PrimaryLine's abbreviated name
- Use SP1 Service to call all numbers that start with **1 and subsequent digits matching SP1 Service's DigitMap. Remove the **1 prefix from the resulting number before making the call
- Use SP2 Service to call all numbers that start with **2 and subsequent digits matching SP2 Service's DigitMap. Remove the **2 prefix from the resulting number before making the call
- Use the LINE port to call all numbers that start with **8 and subsequent digits matching LINE port's DigitMap. Remove the **8 prefix from the resulting number before making the call
- Use the OBiTALK Service to call all numbers that start with **9 and subsequent digits matching OBiTALK Service's DigitMap. Remove the **9 prefix from the resulting number before making the call

Digit Map Configuration

Digit Map Rules and Elements

A digit map serves to transform and restrict the number that may be dialed or called, and determine if sufficient digits have been dialed by the user to form a complete number. Each map is composed of one more rules surrounded by parentheses (which MUST NOT be omitted). Here is the general format of a digit map:

(rule|rule|….|rule)

A digit map *rule* is a rule for matching a given sequence of digits. It may contain extra white spaces for readability; all spaces are removed by the OBi device during parsing. A rule may contain one or more of the following *elements*:

- literals - Any combination of 0-9,*,#,+,-,A-Z,a-z, except m, M, s, S, x, X which have special meaning in the digit map syntax. It matches digit sequences with exactly the same literals
- 'literals' - Everything inside a pair of single quotes is treated as a literal except for the single quote (') character.
- x - a wild card digit that matches any digit from 0-9. x is CASE SENSITIVE
- x. - matches 0 or more x
- [123-7] or [135] - A set of 1 or more digits surrounded by pair of [ ]. It matches any digit in the set. The – syntax represents an inclusive digit range, such as 0–9, 3–7. So [123–7] is equivalent to [1–7] or [1234567]
- S, S0, S1, S2, …S9 - Digit timer of 0, 1, 2, …,9 seconds respectively; S is equivalent to S1; S0 is the same as "blank". You can concatenate multiple S elements together if you need more than 9s timeout, such as S9S5 for a 14s timeout. S is CASE SENSITIVE. It should only be used either as the first element of a rule for hot/warm line implementation, or as the last element of a rule as a means of overriding the default interdigit timer
- <elements:literals> - Substitute the digit sequence matching *elements* with the given *literals*. Single quote syntax is NOT needed or allowed for the *literals* in this context; special characters may be used here as they do not apply in this context either. Elements can be empty, in which case the ':' may be omitted. This case is useful for inserting some extra digits in certain part of the dialed digits. The literals part can be empty also but the ':' MUST NOT be omitted. This case is useful for removing part of dialed digits. *Elements* and *literals* MUST NOT be both empty.
- (*map*) – An embedded digit map for matching subsequent digits.
- (M*label*) – A named embedded digit map for matching subsequent digits, where *label* is one of abbreviated terminal names. Possible choices are:
    - (Msp1) for **SP1 Service::**DigitMap
    - (Msp2) for **SP2 Service::**DigitMap
    - (Msp3) for **SP3**Error! Bookmark not defined. **Service::**DigitMap
    - (Msp4) for **SP4**Error! Bookmark not defined. **Service::**DigitMap
    - (Mpp) for **OBiTALK Service::**DigitMap
    - (Mli) for **LINE Port**Error! Bookmark not defined.**::**DigitMap
    - (Mph) for **PHONE Port::**DigitMap
    - (Mph2) for **PHONE2 Port**Error! Bookmark not defined.**::**DigitMap
    - (Maa) for **Auto Attendant::**DigitMap
    - (Mtg*N*) for TrunkGroup*N*::DigitMap, *N*=1,2,3,4
    - (Mvg*N*) for VoiceGateway*N*::DigitMap, *N*=1,2,3,4,5,6,7,8

Starting with release 1.2, the following elements are added:
- X – A wildcard digit that matches 0–9 or *. This is equivalent to [x*] or [0-9*x]
- @ – A wildcard character that matches any alphanumeric character except #

- x? – matches 0 or 1 x
- @? – matches 0 or 1 @
- [^...] – matches any single alphanumeric character that is not in the set
- Allow alphanumeric and wildcard inside a set [ ], such as [x], [X#], [@#], [a-zA-Zx]

The last two elements imply that the OBi digit maps are *recursive*. Recursive digit maps allow digit maps to be reused and make their specification more compact and readable. It is important that you do not specify digit maps that lead to infinite recursion. For example, ***a digit map must not include a named embedded digit map that references itself***.

To bar users from calling numbers that match a rule, add a '!' in front of that rule in the digit map. The rule is then referred to as a *barring rule*.

Let's look at some examples.

1408xxxxxxx – Matches any 11-digit number that starts with 1408

011xx. – Matches any number that starts with 011 followed by one or more digits

<1408>xxxxxxx – Matches any 7-digit number. OBi pre-pends 1408 to the number when making the call

<:1408>xxxxxxx – Equivalent to the last example

<+>1xxxxxxxxxx – Pre-pends '+' to any 11-digit number that starts with 1

<**1:>1408xxxxxxx – Matches any number that starts with **11408 followed by 7 digits. OBi removes the **1 prefix when making the call

*74(x|xx) – Matches any number that starts with *74, followed by 1 or 2 digits

**1(Msp1) – Matches any number that starts with **1 and with the rest of digits matching the `DigitMap` in the SP1 Service

<:1234> – Matches an empty phone number and replaces with 1234. This is the syntax for a hotline to 1234

<S0:1234> – Equivalent to the last example

<:#> – Hotline to the number #

<S0:#> – Equivalent to the last example

<S4:1234> – Call 1234 if no digits entered for 4s. This is the syntax of a warm line.

xx.853 7683 – Matches any number with at least 8 digits and ends with 8537683, such as 15108537683, 98537683

(x.408 223 1122) – Matches any number with at least 10 digits and ends with 408 223 1122, such as 4082231122, 1408 223 1122

xx.<#> – Adds a # to the end of any number with 1 or more digits

!1900xxx xxxx – Barring all 11-digit numbers that start with 1900

[^*]@@. – Arbitrarily long alphanumeric sequence (except #) that does not start with *

xx? – one or two-digit number

(1xxxxxxxxxxS0|xx.) – Arbitrarily long digit sequence not starting with 1; otherwise it is limited to 11 digits

Matching Against Multiple Rules in Digit Map

One important function of a digit map is to determine if sufficient digits have been entered by the user during dialing. A digit map normally contains more than one rules. The Digit Map Processor (DMP) must return the best matched rule at some point, or declare the input digit sequence is invalid. The DMP keeps refining its decision as each digit is entered until it reaches a *final decision*, or will be forced to make a *timely decision* when the interdigit timer expires.

The DMP restarts the interdigit timer on every newly entered digit. The duration of this timer can be either *long* or *short*. The long and the short timer values are set to 10s and 2s respectively by default and are configurable per phone port via the DigitMapLongTimer and DigitMapShortTimer parameters respectively. Whether to use the long or short interdigit timer depends on the current rule matching states. The DMP maintains a matching state for each rule in the digit map as it processes each input digit. The following states are defined:

- Partially Matched (PM) – The rule partially matches the accumulated input sequence. Initially all rules are in this state before any digit is entered. Rules in this state have the potential of becoming EM or IM as more digits are entered. Example: 1234 partially matches the rules xxxxxxx, 1xxxx, 1234567, <123:>xxxx.
- Exactly Matched (EM) – The rule exactly matches the accumulated input sequence. However, any further input digit will turn this rule into the MM state. Example: 1234 exactly matches the rules xxxx, 1234, 1xxx, <123:5678>x
- Indefinitely Matched (IM) – The rule matches the accumulated input sequence indefinitely, with a variable length such that the rule can potentially stay as IM as more matching digits are entered. Example: 011853 indefinitely matches the rules xx., 011xx., <011:>xx.
- Mismatch (MM) – The rule does not match the accumulated input sequence. This state will not change as more digits are entered. Example: 1234 mismatches the rules 123, 1xx, 12345

Rules in the EM or IM state are candidates to be selected by the DMP. After processing a new digit, the DMP returns a final decision if any of the following conditions holds:

1. All rules are the MM state. DMP returns an error
2. One or more rules are in the EM state with no rules in the IM state. DMP returns the best matched EM rule. If the best matched rule is a barring rule, DMP returns an error instead

Otherwise, DMP starts the short interdigit timer if there is at least one rule in the EM state, or else the long one. When the interdigit timer expires, DMP makes a timely decision by returning the best matched rule at that moment if one is found, or else a timeout error. Again if the best matched rule in this case is a barring rule, DMP returns an error instead. Note that the timer to wait for the first input digit is NOT governed by the interdigit timer, but the duration of dial tone being played and could be a lot lengthier than the long interdigit timer.

The best matched rule is the one that has the most specific literals matching the input digit sequence. For example, the input sequence 1234 matches the rule 123x better than 1xxx. On the other hand, an EM rule is always selected over an IM rule.

Finally, the default interdigit timer can be overridden by appending the S*n* element at the end of the rule (*n* = 0–9).

Let's look at some examples. Consider this simple digit map:

(<1408>xxx xxxx)

As soon as 7 digit have been entered, the DMP returns a complete number by pre-pending the accumulated digits with 1408.

Consider another simple map:

(xx.)

After user dials one or more digits, the DMP returns the accumulated digits as a complete number when the long interdigit timer expires.

Let's combine the last two maps:

(xx. | <1408>xxx xxxx)

After user dials 1 or more digits but less than 7 digits, the DMP would return the accumulated digits as a complete number when the (long) interdigit timer expires. As soon as 7 digits are entered, the DMP would return 1408 followed by the accumulated 7-digit when the (short) interdigit expires. On the 8th digit and beyond, however, the DMP will consider the first rule only and return the accumulated digits as is when the (long) interdigit timer expires.

Now add a S4 timer to the 2nd rule:

(xx. | <1408>xxx xxxxS4)

In this case the DMP behaves exactly the same as the last, except that the short interdigit timer the DMP uses upon receiving the 7th digit is overridden by a 4s timer; hence the user will have up to 4s instead of 2 to dial the 8th digit.

Forcing Interdigit Timeout With A Pound(#) Key

When dialing, user may force an interdigit timeout with a # key instead of waiting for the DMP to timeout its own long or short timer. This is allowed as long as the # key does not match the current element of any PM rules. Otherwise the # key will be "swallowed" by the DMP instead of triggering a timeout.

Consider the digit map (33xx.)

If the user enters 333#, the DMP will return immediately with the number 333.

Now consider the digit map (33xx.|333#1234x.)

If the user enters 333#, the DMP will not return but continue to wait for further input or its interdigit timer to expire. Note that the first rule "33xx." is now in the MM state since the digit # does not match "x". The user may continue to enter 1234#, or 1234 and wait for a long interdigit timeout for the DMP to successfully return 333#1234.

Invoke Second Dial Tone in Digit Map

You can tell OBi to start a tone after a certain pattern of digits have been dialed by specifying the element {t=<tone>} within a digit map, where <tone> is a 1 to 3-letter name of the tone to play. The tone will stop when the next digit is entered. For example:

(**1{t=di2}(Msp)|**8{t=od}(Mli))

which tells the device to play Second Dial Tone when **1 is dialed, or play Outside Dial Tone when **8 is dialed. Here is a full list of acceptable (case insensitive) values of <tone>:

bu = Busy Tone

cf = Call Forwarded Dial Tone

cm = Confirmation Tone

co = Conference Tone

cw1 – cw10 = Call Waiting Tone 1-10, respectively

di = Dial Tone

di2 = Second Dial Tone

fb = Fast Busy Tone

ho = Holding Tone

od = Outside Dial Tone

pr = Prompt Tone

rb = Ringback Tone

ro = Reorder Tone (same as fast busy)

si1 – si4 = SIT TONE 1 – 4, respectively

st = Stutter Tone

0 – 9, *, #, a – d = DTMF 0 – 9, *, #, A – D respectively

Change Inter-digit Long Timer Dynamically After Partial Match

The OBi starts off with the inter-digit long timer set to the configured DigitMapLongTimer value when processing a new digit sequence by a digit map. You may change the long timer as some patterns are partially matched by embedding the syntax {L=<time>} within a rule in the digit map, where <time> is the desired number of seconds for the long timer. For example:

(011 853 xxxx xxxx{L=5}x. |xx.)

Here the long timer is shortened to 5s after the user has entered 011 853 + 8 digits. Hence the OBi will declare that a complete number is collected in 5s when no more digits are received. Without the {L=5} syntax the user will have to wait for 10s (by default) for the same to happen.

User Defined Digit Maps

There are 10 user definable digit maps available under the User Settings – User Defined Digit Maps section of the device configuration web page. These digit maps are referred to as User Defined Digit Map 1 to 10. Each user defined digit map is specified with 2 parameters:

- Label: An arbitrary string for referencing this digit map in other digit map specification. The value should be 2-16 characters long. For example, "friends". In this case, (Mfriends) can be referenced in other digit maps, such as PHONE Port::DigitMap

- DigitMap

By default both parameters are empty, except for User Defined Digit Map 1 (see the section below).

A User Defined Digit Map For IPv4 Dialing

The default values of the parameters for User Defined Digit Map 1 are set the following values to support IPv4 Dialing:

- Label: ipd

- Digit Map: (xx.<*:@>xx?x?<*:.>xx?x?<*:.>xx?x?<*:.>xx?x?|

    xx.<*:@>xx?x?<*:.>xx?x?<*:.>xx?x?<*:.>xx?x?<*::>xx?x?x?x?)

The map (Mipd) is referenced in the default setting of the DigitMap in ITSP Profile A and B. It supports the following two forms of IPv4 dialing:

a) <user-id>*<a>*<b>*<c>*<d>

b) <user-id>*<a>*<b>*<c>*<d>*<port>

where <user-id> is an arbitrary length numeric user-id, such as 100345, <port> is a port number in the range 0–65535, and each of <a>,<b>,<c>,<d> is a 1-3 digit pattern in the range 1–255 that identifies one byte of an IP address. The dialed number will be translated into <user-id>@<a>.<b>.<c>.<d> and <user-id>@<a>.<b>.<c>.<d>:<port> respectively. Here are some examples:

    1234*192*168*15*113   maps to 1234@192.168.15.113

    123456*192*168*15*180*5061   maps to 123456@192.168.15.180:5061

Hunt Groups

A Hunt Group is a list of terminal-sets that should be alerted for an incoming call. The terminal-sets are alerted sequentially one-at-a-time. A terminal-set may include one or more terminals that are alerted simultaneously when the terminal-set is alerted. A phone port PH*n* and SP*n*(*number*), *n* = 1, 2, 3, … are examples of a terminal.

Up to 10 hunt groups can be defined in an OBi5xx device, using the parameters: **Hunt Groups**::**Group***n*, for *n* = 1, 2, … 10. The device can be configured to route certain incoming calls to be handled by one of the hunt groups, using the inbound call route parameters, in a way that is no different from routing incoming calls to ring any one of the phone ports.

In order to hunt sequentially for an answer, each terminal-set should be set to ring for a finite duration say x seconds, such that if none of the terminals answer the call within x seconds, the device moves on to ring the next terminal-set or stops the hunt if there are no more terminal-sets to try. In the case when all the terminals in a set are busy, the device will skip over the set and moves on immediately. On the other hand, if the intention of the hunt is to find the first *terminal-set* that is not busy, it is perfectly acceptable to specify in the hunt group configuration to ring each *terminal-set* indefinitely.

The general format of a hunt group parameter value is (all case-insensitive):

GP (*name*) = { *ringlist* } ; *group-par* ; *group-par* ; …

where

- *name* = arbitrary string with no spaces for referencing this hunt group. It can be a number such as 1 or a generic string such as sales

- *ringlist* is made up of one or more *terminal-set* separated by a +, with the following general format:

  o *ringlist = terminal-set +terminal-set +*….

  The *terminal-sets* are alerted sequentially one-at-a-time.

- *group-par* is a parameter to control how to alert the members in the *ringlist*. It has the format:

  o *group-parameter-name=value*

  o The following *group-parameter-names* and corresponding *values* are defined:

    ▪ du: *value* is the (default) duration in number of seconds to alert the *terminal-set* in each pass or * to alert each set indefinitely

    ▪ na: *value* is the terminal or hunt group to alert if the call is not answered by any terminal in the hunt group; this mechanism may be used to daisy-chain hunt groups

    ▪ np: *value* is the number of passes to cycle through the *ringlist*. Default is 1

- A *terminal-set* is made up of a comma-separated list of *terminals* that are alerted simultaneously when the *terminal-set* is alerted. It has the following general format:

    terminal-set = terminal,terminal,…;set-par;set-par;…

  where

  o *set-par* is a parameter that controls the *terminal-set*. It has the format:

      set-parameter-name=value

    The following *set-parameter-names* and corresponding *values* are defined:

- du: *value* is the duration in number of seconds to alert the *terminal-set*, or * to alert the set indefinitely. If this parameter is not specified for the set, it takes the value from the group's du parameter if one is specified. Otherwise the set will ring indefinitely

Tips: You may sprinkle the hunt group parameter value with spaces to make it more readable. Beware that all white spaces will be removed prior to processing these parameter values (including the ones in the hunt group *name*)

Examples:

1. Ring ph1,…,ph8 simultaneously

   `GP(support)={ph1,ph2,ph3,ph4,ph7,ph8}`

2. Ring ph1, p2, ph3 sequentially; the first phone port that rings grabs the call. If all phones are busy, ring 14089991234 (via SP1 Service):

   `GP(1)={ph1+ph2+ph3};na=sp1(14089991234)`

3. Ring each of ph1, ph2, and ph3 sequentially for 20s, twice, then forwards the call to 14089991123 if no one answers:

   `GP(2)={ph1+ph2+ph3};dur=20;np=0;na=sp1(14089991123)`

4. Ring ph1,p2,ph3 simultaneously for 30s, then ring ph4 and ph5 simultaneously for 30s, then forward the call to the hunt group named general if no one answers:

   `GP(sales)={ph1,ph2,ph3+ph4,ph5;du=30+ph6};du=20;na=gp(general)`

5. Same as last one but also override the na and du group parameters of the hunt group named general:

   `GP(sales)={ph1,ph2,ph3+ph4,ph5;du=30+ph6};du=20;na=gp(general;du=10;na=sp1(14089991234))`

Routing Incoming Calls to Hunt Groups

By adding rules to an inbound call route parameter, you can send incoming calls to be handled by a hunt group. For example, add the rules: `{>18007771000:gp(sales)},{gp(general)}` to **SP***n* **Service**::**X_InboundCallRoute** parameter, such that all calls to 18007771000 will be handled by the hunt group named sales, and the other calls by the hunt group named general.

Hunt Groups Web Page

Parameter	Description	Type	Access	Default Setting
Hunt Groups (**VoiceService.1.X_HuntGroups.**)				
Group*n* (Repeat for *n* = 1 – 10)	A hunt group specification	String	RW	
VirtualStacking	Enable Virtual Stacking operation to allow hunting terminals on external OBi5xx units	Bool	RW	`false`

SIP Trunking

There are 9 ITSP Profiles and 9 SP Services available in each OBi500 device. Each SP Service can be configured as a SIP Trunk in the following sense:

1. Single REGISTRATION per SP account

2. Multiple DID numbers for incoming calls and outgoing calls

3. Incoming call routing based on called DID numbers, to ring any combinations of phone ports simultaneously or a hunt group

4. Static Call Forwarding according to DID numbers via hunt groups

5. Outgoing caller identity based on the phone port that originates the call

6. Option to insert user=phone for caller identity (for a DID number) in outgoing calls

Configuration of a SIP Trunk on SP1 (ITSP Profile A) on OBi508

In this example, we use SP1 on ITSP Profile A as SIP trunk.

Use Case:

- All 8 phone ports use SP1 for outgoing calls

- Each phone port has a DID number, +14089991001, +14089991002, … +14089991008, respectively

- Let *mytrunkid* be the Pilot UserID of the SIP trunk

Configuration:

Parameter	Value	Description
ITSP Profile A–SIP:: **X_SpoofCallerID**	`true`	Enable this option to let the device insert a UserID in the FROM header of outbound INVITE that is different from the SIP Account UserID
ITSP Profile A–SIP:: **X_UserEqPhone**	`true`	Enable this option to let the device insert a **;user=phone** (URI) parameter in the Request-URI and TO header of outbound INVITE (for all calls on SP1). Do not enable this option if the server does not expect to see this URI parameter
SP 1 Service:: **AuthUserName**	`mytrunkid`	Configure the SIP Account credentials. If the UserID for authentication is different from the SIP UserID, the SIP UserID must be configured in the **URI** parameter and authentication UserID in the **AuthUserName** parameter.
SP1 Service:: **AuthPassword**	The authentication password corresponds to mytrunkid	
SP1 Service:: **URI**		
SP1 Service:: **X_InboundCallRoute**	`{>(@.4089991001):ph1},` `{>(@.4089991002):ph2},…,` `{>(@.4089991008):ph8}`	Configure incoming calls to a specific DID number to ring the corresponding phone port only. The @. syntax matches any number of characters before the last 10 digits of the called number (for example: + or +1 or +001). Note that you can also ring any combination of the phone ports for a

		given called number, or ring a hunt group (GP) defined in the device configuration
Phone 1 Port:: **OutboundCallRoute**	`{(Mpli):pli(+14089991001;context=abd.com;;user=phone>$2)}`	Calls made from each phone port to assume a different identify based on assigned DID number. This identify is shown as the UserID in the From header (and Remote-Party-ID header if enabled) of the outbound INVITE. The Contact header of the INVITE on the other hand uses the Pilot UserID (the SP Account USERID) as the UserID. `$2` represents the target number that the user dials. Note that additional call routing rules may be added for other call destinations. Note that UserID parameters after the first `;` in the UserID are included as is as part of the user id, but parameters after the first `;;` are inserted as URI parameters.
Phone 2 Port:: **OutboundCallRoute**	`{(Mpli):pli(+14089991002>$2)}`	
...	`{(Mpli):pli(+1408999100n>$2)}` where n = 1, 2, …, 8	
Phone 8 Port:: **OutboundCallRoute**	`{(Mpli):pli(+14089991008>$2)}`	
Phone 1 Port:: **PrimaryLine**	`SP1 Service`	All phone ports to use SP1 as the primary line (pli).
Phone 2 Port:: **PrimaryLine**	`SP1 Service`	
...	`SP1 Service`	
Phone 8 Port:: **PrimaryLine**	`SP1 Service`	

Notes:

- You can tell the device to insert a `user=phone` URI parameter in the FROM header of outbound INVITE (and Remote-Party-ID header if enabled) when spoofing the calling UserID with a DID number, by including a `;;user=phone` parameter after the spoofed userid in an OutboundCallRoute rule. In fact, everything in the rule's argument after the first `;;` is included as URI parameter(s).

Call Forward Settings Per Phone Port

There is a set of call forward settings per phone port, as shown in the following table:

Parameter Phone Port n – Call Forwarding::	Description
ForwardAll	Enable forwarding all calls unconditionally. Value must be `true` or `false`.
ForwardAllNumber	The number to forward all calls to unconditionally. It may include the Caller-ID number to use as the UserID of the SIP/Diversion header. Examples: `SP1(1000)` `SP1(+14089991003>1000)` `SP1(+14089991003;;user=phone>1000)` `SP1(+14089991003;context=test.com;;user=phone>1000)`
ForwardOnBusy	Enable forwarding calls when the phone port is busy. Value must be `true` or `false`.
ForwardOnBusyNumber	The number to forward calls to when the phone port is busy. See

	ForwardAllNumber for examples.
ForwardOnNoAnswer	Enable forwarding calls when the call is not answered after a few rings. Value must be `true` or `false`.
ForwardOnNoAnswerNumber	The number to forward calls to when the call is not answered after a few rings. See ForwardAllNumber for examples.
ForwardOnNoAnswerRingCount	The number of rings to wait before triggering call forwarding on no answer, where one ring is equivalent to six seconds

Note that call forwarding from a phone port is implemented only by **bridging** two calls together by the device. That is, the device maintains the call with the original incoming caller (in the ringing state), while making a new call to the call forward target number. When the target rings or answers (whichever happens first), it bridges the two call legs together. The SIP/INVITE to the call forward target will have the original caller's Caller ID in the FROM header, and the Request-UserID of the original SIP/INVITE in a **DIVERSION** header. If the original Request-UserID has a `user=phone` parameter, the DIVERSION header will carry the same URI parameter as well.

DND Setting Per Phone Port

There is a DND (Do Not Disturb) setting per phone port, as shown in the following table:

Parameter Phone Port *n* – Calling Features::	Description
DoNotDisturbEnable	Enable Do Not Disturb on a phone port. Value must be `true` or `false`.

Using Star Codes to Change Per-Phone-Port Call Forward and DND Settings

Assuming a different StarCodeProfile is used for each phone port, one can define local star codes to enable/disable the call forward settings per phone port, as shown below:

For Phone Port *n* (*n* = 1, 2, 3, …, 8) using Star Code Profile *x* (*x* = A, B, …. H)

```
*72, Cfwd All, coll(PHn($Cfan)), set(PHn($Cfa),1)

*73, Disable Cfwd All, set(PHn($Cfa), 0)

*60, Cfwd Busy, coll(PHn($Cfbn)), set(PHn($Cfb),1)

*61, Disable Cfwd Busy, set(PHn($Cfb), 0)

*62, Cfwd No Ans, coll(PHn($Cfnn)), set(PHn($Cfn),1)

*63, Disable Cfwd No Ans, set(PHn($Cfn),0)

*78, Enable DND, set(PHn($Dnd),1)

*79, Disable DND, set(PHn($Dnd),0)
```

It is also possible to use the same star code profile to change the values of call forwarding and DND settings for each phone port, with the generic phone port versions of corresponding star code variables; the phone port whose settings to be changed in this case is the one that the star code is dialed from:

```
*72, Cfwd All, coll($PhCfan), set($PhCfa,1)
```

```
*73, Disable Cfwd All, set($PhCfa, 0)

*60, Cfwd Busy, coll($PhCfbn), set($PhCfb,1)

*61, Disable Cfwd Busy, set($PhCfb, 0)

*62, Cfwd No Ans, coll($PhCfnn), set($PhCfn,1)

*63, Disable Cfwd No Ans, set($PhCfn,0)

*78, Enable DND, set($PhDnd,1)

*79, Disable DND, set($PhDnd,0)
```

MWI Notification Routing

Your OBi device can handle SIP/NOTIFY for MWI status based on the DID number in the Request-UserID of the NOTIFY request. The same processing is used inside or outside the context of a Subscription dialog. You can define routing rules to process the MWI status by any combination of phone ports. There are two settings for this under each **SPn Service–Calling Features**::**X_MWIRoute** for basic MWI handling (by playing Stutter Tone) and **SPn Service–Calling Features**::**X_VMWIRoute** for VMWI. Here is an example of a MWI Route:

`{(@.4089991001)>ph1},{(@.4089991002)>ph2},…,{(@.4089991008)>ph8},{ph7,ph8}`

In the last example, SIP/NOTIFY sent for a particular DID number is handled by a corresponding phone port only, except the last rule which is a catch-all rule for SIP/NOTIFY sent for anything else.

Virtual Stacking

Virtual Stacking is a set of configuration parameters that links a few OBi5xx devices to provide more phone ports in the context of Hunt Groups and SIP Trunking. The linking of the devices can be achieved on the same LAN or across the Internet. OBi5xx supports virtually stacking up to 4 devices to provide 32 phone ports. One of the devices must be designated as the Master Unit (MU), and the other devices are called Expansion Units (EU): EU1 for the first EU, EU2 for the second, and EU3 for the third.

Note that any OBi5xx Unit can be MU or EU. Given 4 OBi500 devices to form a virtual stack, we recommend to use the convention to assign devices as MU and EU based on their OBi numbers, such that the device with the smallest OBi number is the MU, the next smallest EU1, the next smallest EU2, and so on.

There are two applications that virtual stacking is particularly useful:

- Forming hunt groups than span up to 32 phone ports

- SIP trunking with up to 32 DID numbers to support corresponding number of analog phone lines

We will discuss the device configuration for these applications in the following sections. With no loss of generality, let's assume the MU has a SIP trunk installed on SP1. On the MU, you must enable the Boolean parameter **Hunt Groups**::**VirtualStacking**.

Case 1: All Units Having Static or Reserved IP Addresses On the Same Subnet

Let the MU, EU1, EU2, and EU3 have static or reserved IP address *ipaddr0, ipaddr1, ipadd2*, and *ipaddr3* respectively, on the *same subnet*. On the MU, pick any unused SPn Service (n = 2, 3,…) to communicate with the EU. On the EU, pick any unused SPn Service (n = 1, 2, 3, …) to communicate with the MU. We use SP2 on the MU and SP1 on the EU in the current example.

Extending Hunt Groups on the MU to span 32 Phone Ports

On the MU, use the following terminals in the configuration of hunt groups

- `SP2($1>n@ipaddr1)` , n = 1009-1016 to hunt, respectively, PH1-PH8 ports on EU1

- `SP2($1>n@ipaddr2)` , n = 1017-1024 to hunt, respectively, PH1-PH8 ports on EU2

- `SP2($1>n@ipaddr3)` , n = 1025-1032 to hunt, respectively, PH1-PH8 ports on EU3

On EU1, set **SP1 Service**::**X_InboundCallRoute** = `{>1009:ph1},{>1010:ph2},… {>1016:ph8}`

On EU2, set **SP1 Service**::**X_InboundCallRoute** = `{>1017:ph1},{>1018:ph2},… {>1024:ph8}`

On EU3, set **SP1 Service**::**X_InboundCallRoute** = `{>1025:ph1},{>1026:ph2},… {>1032:ph8}`

Extending SIP Trunking on the MU to support 32 DID Numbers

Let the DID numbers be: +14089991001, +14089991002, … +14089991032, such that that:

+14089991001 – +14089991008 are assigned to PH1-PH8, respectively, on the MU

+14089991009 – +14089991016 are assigned to PH1-PH8, respectively, on the EU1

+14089991017 – +14089991024 are assigned to PH1-PH8, respectively, on the EU2

+14089991001 – +14089991008 are assigned to PH1-PH8, respectively, on the EU3.

On EU1:

- Set **SP1 Service**::**X_InboundCallRoute** = `{>1009:ph1},{>1010:ph2},… {>1016:ph8}`

- Set **Phone** n **Port**::**OutboundCallRoute** =`{sp1(`m`>$2)}` for n = 1–8 corresponding to m = 1009–1016, respectively

On EU2:

- Set **SP1 Service**::**X_InboundCallRoute** = `{>1017:ph1},{>1018:ph2},… {>1024:ph8}`

- Set **Phone** n **Port**::**OutboundCallRoute** =`{sp1(`m`>$2)}` for n = 1–8 corresponding to m = 1017–1024, respectively

On EU3:

- Set **SP1 Service**::**X_InboundCallRoute** = `{>1025:ph1},{>1026:ph2},… {>1032:ph8}`

- Set **Phone** n **Port**::**OutboundCallRoute** =`{sp1(`m`>$2)}` for n = 1–8 corresponding to m = 1025–1032, respectively

On the MU:

- Set **ITSP Profile A–SIP**::**X_SpoofCallerID** = `true`

- Set **SP1 Service**::**X_InboundCallRoute** =

```
{>(@.4089991001):ph1},
{>(@.4089991002):ph2},
…,
{>(@.4089991008):ph8},
{>(@.4089991009):sp2($1>1009@ipaddr1)},
{>(@.4089991010):sp2($1>1010@ipaddr1)},
…,
{>(@.4089991016):sp2($1>1016@ipaddr1)},
{>(@.4089991017):sp2($1>1017@ipaddr2)},
{>(@.4089991018):sp2($1>1018@ipaddr2)},
…,
{>(@.4089991024):sp2($1>1024@ipaddr2)},
{>(@.4089991025):sp2($1>1025@ipaddr3)},
{>(@.4089991026):sp2($1>1026@ipaddr3)},
…,
{>(@.4089991032):sp2($1>1032@ipaddr3)}
```

- Set **Phone** n **Port**::**OutboundCallRoute** =

```
{(Msp1):sp1(+1408999100n;context=xyz.com;;user=phone>$2)}
```

for n = 1, 2, …, 8

- Set **SP2 Service**::**X_InboundCallRoute** =

```
{1009:sp1(+14089991009;context=xyz.com;;user=phone>$2)},
{1010:sp1(+14089991010;context=xyz.com;;user=phone>$2)},
…,
{1032:sp1(+14089991032;context=xyz.com;;user=phone>$2)},
```

On EU1, EU2, and EU3:

- Set **Phone *n* Port**::**PrimaryLine** = `SP1 Service` for all phone ports
- Set **ITSP Profile A**::**ProxyServer** = *ipaddr0*
- Set **ITSP Profile A**::**ProxyServerPort** = `5061` (Note: 5061 is the default **SP2 Service**::**X_UserAgent Port**)
- Set **ITSP Profile A–SIP**::**X_SpoofCallerID** = `true`
- Set **SP1 Service**::**RegisterEnable** = `false`

Extending SIP Trunking on the MU to support 32 Voicemail Accounts

Continuing from the last example, suppose further that the SIP Trunk has 32 voicemail accounts corresponding to the 32 DID numbers, with the voicemail account ID *mboxn*, for *n* = 1, 2, … 32. When the voicemail server sends a MWI (a SIP/**NOTIFY** with **Event:message-summary**) to the device, it may indicate the mailbox account ID as the User ID field in the request URI of the SIP/NOTIFY, or in the Message-Account field in the message's body. The OBi5xx looks for the ID in the **Message-Account** field first. If this is not found, it takes the ID from the request URI. The OBi5xx routes the MWI message according to the **X_MWIRoute** parameter for the current SIP trunk (SP1 Service in this example), which may be extended to include routing the message to EUs. Below is an example that routes MWI message based on one-to-one mapping of voicemail account to phone port.

On the MU:

- Set **SP1 Service**::**X_MWIRoute** =

   {*mbox1*:ph1},{*mbox2*:ph2},…,{*mbox8*:ph8},

   {*mbox9*:sp2(1009@*ipaddr1*)},{*mbox10*:sp2(1010@*ipaddr1*)},...,{*mbox16*:sp2(1016@*ipaddr1*)},

   {*mbox17*:sp2(1017@*ipaddr2*)},{*mbox18*:sp2(1018@*ipaddr2*)},...,{*mbox24*:sp2(1024@*ipaddr2*)},

   {*mbox25*:sp2(1025@*ipaddr3*)},{*mbox26*:sp2(1026@*ipaddr3*)},...,{*mbox32*:sp2(1032@*ipaddr3*)}

- Set **SP1 Service**::**X_VMWIRoute** =

   {*mbox1*:ph1},{*mbox2*:ph2},…,{*mbox8*:ph8},

ON EU1:

- Set **SP1 Service**::**X_MWIRoute** = {1009:ph1},{1010:ph2},…,{1016:ph8}

- Set **SP1 Service**::**X_VMWIRoute** = {1009:ph1},{1010:ph2},…,{1016:ph8}

ON EU2:

- Set **SP1 Service**::**X_MWIRoute** = {1017:ph1},{1018:ph2},…,{1024:ph8}

- Set **SP1 Service**::**X_VMWIRoute** = {1017:ph1},{1018:ph2},…,{1024:ph8}

ON EU3:

- Set **SP1 Service**::**X_MWIRoute** = {1025:ph1},{1026:ph2},…,{1032:ph8}

- Set **SP1 Service**::**X_VMWIRoute** = {1025:ph1},{1026:ph2},…,{1032:ph8}

A More Specific Example:

Given 4 OBi508 where the MU is configured as a router, and the 3 EU are connected to the LAN ports of the MU as shown in the figure below:
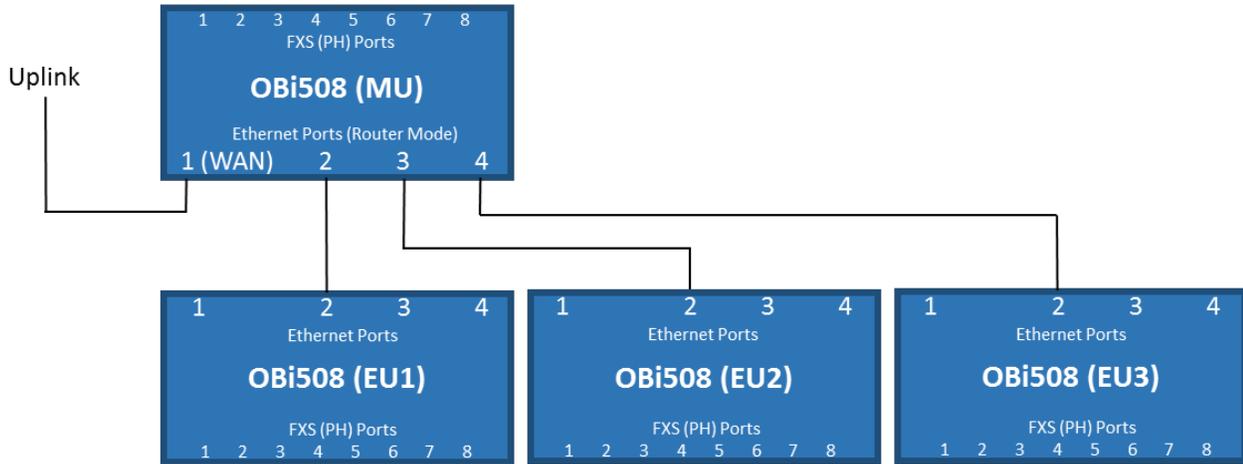
All EU devices have DHCP enabled and therefore will have IP addresses assigned by the DHCP server in the MU. Let *mac1*, *mac2*, and *mac3* be the MAC addresses of EU1, EU2, and EU3 respectively. We can reserve the IP addresses to be assigned to these 3 MAC on the MU. Hence, in referencing to the last example, *ipaddr0* would the same as the value of **LAN Settings**::**RouterIPAddress** on the MU, and *ipaddr1*, *ipaddr2*, and *ipaddr3* would be equal to the addresses reserved for *mac1*, *mac2*, and *mac3* respectively on the MU.

Case 2: MU/EU Located Anywhere On the Internet

Let the MU, EU1, EU2, and EU3 have OBi Numbers $obin0$, $obin1$, $obn2$, and $obn3$ respectively, and reachable to OBiTALK.com via the Internet.
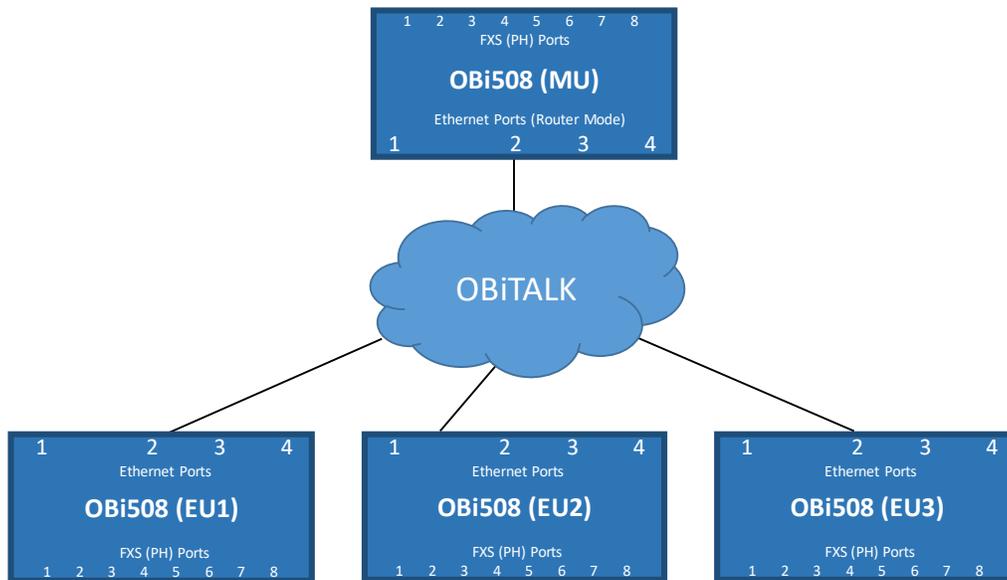
Extending Hunt Groups on the MU to span 32 Phone Ports

On the MU, use the following terminals in the configuration of hunt groups

- `VG1(n)`, *n* = 1009-1016 to hunt, respectively, PH1-PH8 ports on EU1

- `VG2(n)`, *n* = 1017-1024 to hunt, respectively, PH1-PH8 ports on EU2

- `VG3(n)`, *n* = 1025-1032 to hunt, respectively, PH1-PH8 ports on EU3

- **Voice Gateway1**::**AccessNumber** = *obn1*

- **Voice Gateway2**::**AccessNumber** = *obn2*

- **Voice Gateway3**::**AccessNumber** = *obn3*

- **Voice Gateway1**::**SpoofCallerID** = `true`

- **Voice Gateway2**::**SpoofCallerID** = `true`

- **Voice Gateway3**::**SpoofCallerID** = `true`

On EU1, set **OBiTALK Service**::**X_InboundCallRoute** = `{>1009:ph1},{>1010:ph2},… {>1016:ph8}`

On EU2, set **OBiTALK Service**::**X_InboundCallRoute** = `{>1017:ph1},{>1018:ph2},… {>1024:ph8}`

On EU3, set **OBiTALK Service**::**X_InboundCallRoute** = `{>1025:ph1},{>1026:ph2},… {>1032:ph8}`

Extending SIP Trunking on the MU to support 32 DID Numbers

Let the DID numbers be: +14089991001, +14089991002, … +14089991032, such that that:

> +14089991001 – +14089991008 are assigned to PH1-PH8, respectively, on the MU
>
> +14089991009 – +14089991016 are assigned to PH1-PH8, respectively, on the EU1
>
> +14089991017 – +14089991024 are assigned to PH1-PH8, respectively, on the EU2
>
> +14089991001 – +14089991008 are assigned to PH1-PH8, respectively, on the EU3.

On EU1:

- Set **OBiTALK Service**::**X_InboundCallRoute** = {*obn0*>1009:ph1},{*obn0*>1010:ph2},… {*obn0*>1016:ph8}

- Set **Phone** *n* **Port**::**OutboundCallRoute** = {vg1(*m*>$2)} for *n* = 1–8 corresponding to *m* = 1009–1016, respectively. Alternatively, using PP directly without going VG, set

  Phone *n* **Port**::**OutboundCallRoute** = {PP(*m@obn0*>$2)} for *n* = 1–8 corresponding to *m* = 1009–1016

On EU2:

- Set **OBiTALK Service**::**X_InboundCallRoute** = {*obn0*>1017:ph1},{*obn0*>1018:ph2},… {*obn0*>1024:ph8}

- Set **Phone** *n* **Port**::**OutboundCallRoute** = {vg1(*m*>$2)} for *n* = 1–8 corresponding to *m* = 1017–1024, respectively. Alternatively, using PP directly without going VG, set

  Phone *n* **Port**::**OutboundCallRoute** = {PP(*m@obn0*>$2)} for *n* = 1–8 corresponding to *m* = 1017–1024

On EU3:

- Set **OBiTALK Service**::**X_InboundCallRoute** = {*obn0*>1025:ph1},{*obn0*>1026:ph2},… {*obn0*>1032:ph8}

- Set **Phone** *n* **Port**::**OutboundCallRoute** = {vg1(*m*>$2)} for *n* = 1–8 corresponding to *m* = 1025–1032, respectively. Alternatively, using PP directly without going VG, set

  Phone *n* **Port**::**OutboundCallRoute** = {PP(*m@obn0*>$2)} for *n* = 1–8 corresponding to *m* = 1025–1032

On EU1, EU2, and EU3:

- Set **Voice Gateway1**::**AccessNumber** = *obn0*

- Set **Phone** *n* **Port**::**PrimaryLine** = Trunk Group 1 for all phone ports

- Set **Trunk Group 1**::**TrunkList** = vg1

- Set **Voice Gateway 1**::**SpoofCallerID** = true

On the MU:

- Set **SP1 Service**::**X_InboundCallRoute** =

```
{>(@.4089991001):ph1},
{>(@.4089991002):ph2},
```

```
…,
{>(@.4089991008):ph8},
{>(@.4089991009):VG1($1>1009)},
{>(@.4089991010):VG1($1>1010)},
…,
{>(@.4089991016):VG1($1>1016)},
{>(@.4089991017):VG2($1>1017)},
{>(@.4089991018):VG2($1>1018)},
…,
{>(@.4089991024):VG2($1>1024)},
{>(@.4089991025):VG3($1>1025)},
{>(@.4089991026):VG3($1>1026)},
…,
{>(@.4089991032):VG3($1>1032)}
```

Alternatively, you may use PP directly without going through VG, albeit somewhat lengthier:

SP1 Service::**X_InboundCallRoute** =

```
{>(@.4089991001):ph1},
{>(@.4089991002):ph2},
…,
{>(@.4089991008):ph8},
{>(@.4089991009):PP($1>1009@obn1)},
{>(@.4089991010):PP($1>1010@obn1)},
…,
{>(@.4089991016):PP($1>1016@obn1)},
{>(@.4089991017):PP($1>1017@obn2)},
{>(@.4089991018):PP($1>1018@obn2)},
…,
{>(@.4089991024):PP($1>1024@obn2)},
{>(@.4089991025):PP($1>1025@obn3)},
{>(@.4089991026):PP($1>1026@obn3)},
…,
{>(@.4089991032):PP($1>1032@obn3)}
```

- Set **Phone n Port**::**OutboundCallRoute** =

```
{(Msp1):sp1(+1408999m;context=xyz.com;;user=phone>$2)}
```

  for n = 1, 2, …, 8 corresponding to m = 1001 – 1009, respectively

- Set **OBiTALK Service**::**X_InboundCallRoute** =

```
{1009@obn1:sp1(+14089991009;context=xyz.com;;user=phone>$2},
{1010@obn1:sp1(+14089991010;context=xyz.com;;user=phone>$2},
…,
{1016@obn1:sp1(+14089991010;context=xyz.com;;user=phone>$2},
{1017@obn2:sp1(+14089991017;context=xyz.com;;user=phone>$2},
…,
{1024@obn2:sp1(+14089991024;context=xyz.com;;user=phone>$2},
{1025@obn3:sp1(+14089991025;context=xyz.com;;user=phone>$2},
…,
{1032@obn3:sp1(+14089991032;context=xyz.com;;user=phone>$2}
```

Note that for OBiTALK service, caller-id matching is based on the caller's OBi number. To match caller's number, use the syntax *number@OBiNumber* to match both the caller's number and obi number, or *number@* to match just the caller's number, in the caller-id field insider PP() in the inbound call route. Note also that when routing a call to go out from OBiTALK (i.e. PP), the caller number and caller obi number are always spoofed; no setting is required. On the other hand, when routing call to go out via VG, the caller id is not spoofed by default; you must enable the **SpoofCallerID** option on the VG to spoof the caller-id.

Extending SIP Trunking on the MU to support 32 Voicemail Accounts

Continuing from the last example, suppose further that the SIP Trunk has 32 voicemail accounts corresponding to the 32 DID numbers, with the voicemail account ID *mboxn*, for *n* = 1, 2, … 32. When the voicemail server sends a MWI (a SIP/**NOTIFY** with **Event:message-summary**) to the device, it may indicate the mailbox account ID as the User ID field in the request URI of the SIP/NOTIFY, or in the Message-Account field in the message's body. The OBi5xx looks for the ID in the **Message-Account** field first. If this is not found, it takes the ID from the request URI. The OBi5xx routes the MWI message according to the **X_MWIRoute** parameter for the current SIP trunk (SP1 Service in this example), which may be extended to include routing the message to EUs. Below is an example that routes MWI message based on one-to-one mapping of voicemail account to phone port.

On the MU:

- Set **SP1 Service**::**X_MWIRoute** =

    `{`*mbox1*`:ph1},{`*mbox2*`:ph2},…,{`*mbox8*`:ph8},`

    `{`*mbox9*`:pp(1009@`*obn1*`)},{`*mbox10*`:pp(1010@`*obn1*`)},...,{`*mbox16*`:pp(1016@`*obn1*`)},`

    `{`*mbox17*`:pp(1017@`*obn2*`)},{`*mbox18*`:pp(1018@`*obn2*`)},...,{`*mbox24*`:pp(1024@`*obn2*`)},`

    `{`*mbox25*`:pp(1025@`*obn3*`)},{`*mbox26*`:pp(1026@`*obn3*`)},...,{`*mbox32*`:pp(1032@`*obn3*`)}`

- Set **SP1 Service**::**X_VMWIRoute** =

    `{`*mbox1*`:ph1},{`*mbox2*`:ph2},…,{`*mbox8*`:ph8},`

ON EU1:

- Set **OBiTALK Service**::**X_MWIRoute** = `{1009:ph1},{1010:ph2},…,{1016:ph8}`
- Set **OBiTALK Service**::**X_VMWIRoute** = `{1009:ph1},{1010:ph2},…,{1016:ph8}`

ON EU2:

- Set **OBiTALK Service**::**X_MWIRoute** = `{1017:ph1},{1018:ph2},…,{1024:ph8}`
- Set **OBiTALK Service**::**X_VMWIRoute** = `{1017:ph1},{1018:ph2},…,{1024:ph8}`

ON EU3:

- Set **OBiTALK Service**::**X_MWIRoute** = `{1025:ph1},{1026:ph2},…,{1032:ph8}`
- Set **OBiTALK Service**::**X_VMWIRoute** = `{1025:ph1},{1026:ph2},…,{1032:ph8}`

Final Thoughts

We have discussed two specific cases of virtual stacking: a) All devices within the same subnet with static/reserved IP addresses and b) All devices are scattered on the Internet and linked via OBiTALK. In general, the two cases can be mixed where some devices are on the same subnet as the MU that also serves as their router (gateway), and other devices scattered elsewhere on the Internet. The architecture thus allows for example the MU and some EUs in the main office, but have one or more devices in branch offices.

Specifications & Environmental Characteristics of the OBi500 Device

OBi508

General

Brand	Obihai Browse Obihai Devices
Manufacturer	Obihai
Hardware Designer	Obihai Technology, Inc.
Model Name	OBi508
Release Date	January 2014

Microprocessor

Chip Vendor	Obihai SoC
Width of Machine Word	32 bit
Instruction Set	ARM

FXS SLIC (Subscriber Line Integrated Circuit)

Ringer Specifications	Ring Frequency: 14Hz – 68Hz Ring Waveform: Trapezoidal, Sinusoidal Ring Voltage: 55v – 85v
Maximum Ring Load	5 REN (Ringer Equivalence Number)
FXS (PHONE Port) Configuration Settings	Recursive Digit Map & Associated Outbound Call Routing On-Hook Tip Ring Voltage: 30v – 52v Off-Hook Current Max: 15mA – 45mA Impedance: 12 Independent Settings DTMF Playback Level: -90 dBm – 3dBm Caller ID Method (FSK Bell202, FSK V.23, DTMF FI/SE/DK) Caller ID Trigger (Before / After First Ring, Polarity Reversal) Channel Tx Gain: -12dB – 12dB Channel Rx Gain: -12dB – 12dB Silence Detect Sensitivity Hook--Flash Time Max HookFlash Time Min CPC Delay Time CPC Duration Idle Polarity Connect Polarity

Management – Configuration

Local Access Interface	IVR, Web Page – Password Protected (Admin & User Level Log-in)
Remote Access Interface	Syslog (Multi-Level Granularity), Invokable via SIP Notify, Web, Provisioning
Device Web Page Standard	HTTP v1.1, XML v1.0
Remote Provisioning	XML via TFTP or HTTP, TR069 / TR104
Secure Remote Provisioning	SSL via HTTPS , Encrypted XML via HTTP or TFTP – Dedicated User Name & Password
Secure Remote Firmware Update	Encrypted Binary File via TFTP or HTTP + Dedicated User Name & Password
Customization	OBi-ZT: Obihai Zero-Touch Automatic Customization & Configuration **
Call History (CDRs)	Call Detail Records on OBi Web Page, Export to XML
LED Indications	Power, Device Status, Upgrade Progress Status, Ethernet Activity, PHONE Status
RTP Statistics	RTP Transport Type

Management Configuration – Continued…	Audio Codec Type (Tx/Rx)
	RTP Packetization - ms (Tx/Rx)
	RTP Packet Count (Tx/Rx)
	RTP Byte Count (Tx/Rx)
	Peer Clock Differential Rate - PPM
	Packets In Jitter Buffer
	Packets Out-Of-Order
	Packets Interpolated
	Packets Late (Dropped)
	Packets Lost
	Packet Loss Rate %
	Packet Drop Rate %
RTP Statistics Cont…	Jitter Buffer Length - ms
	Received Interarrival Jitter - ms
	DTMF Digits Received
	Jitter Buffer Underruns
	Jitter Buffer Overruns
	Sequence Number Discontinuities
	Skew Compensation - ms
Session Information	SIP Session Status
	OBiTALK Status
	Phone Port Status
Primary SIP Service Set-Up Wizard	Dedicated Device Web Page for Quick ITSP Account Set-Up
System Settings Back-Up / Restore	Save & Restore Configuration via XML file to / from a Local Folder

Security

Local Access Interface	IVR Password
Remote Access Interface	User Name & Password Access via HTTP, TFTP – HTTPS
Device Web Page Standard	HTTP v1.1, XMLv1.0
Secure Remote Provisioning	TFTP, HTTP, HTTPS

Network – Application Details

Data Networking	MAC Address (IEEE 802.3)
	UDP (RFC 768)
	TCP (RFC 793)
	IP version 4 (RFC 791) – Static IP and DHCP Support
	ICMP (RFC 792)
	ARP - Address Resolution Protocol
	RTP (RFC 1889, 1890)
	RTCP (RFC 1889)
	DHCP Client (RFC 2131)
	DiffServ (RFC 2475) – Independently Configured: Service, SIP & Media
	ToS (RFC 791, 1349) – Independently Configured: Service, SIP & Media
	VLAN Tagging (802.1p) – Independently Configured: Service, SIP & Media
	SNTP (RFC 2030) – Primary & Secondary NTP Servers
VoIP	SIPv2 (RFC 3261, 3262, 3263, 3264)
	SIP over UDP
	SIP over TCP
	SIP over TCP with TLS
	8 SIP Service Provider Service Sessions – Concurrent Operation
	8 XMPP (Google Voice) Sessions
	1 OBiTALK Service Session
	SIP Proxy Redundancy – Local or DNS Based SVR, Primary & Secondary Fallback List
	Restrict Source IP Address
	Maximum Number of Sessions – Independent per Service
	Trunk Groups (4)
	Voice Gateway – Direct Dialing (8)

Network – Application Details Continued…	G.711 A-Law
	G.711 μ-Law
	G.726 (@32 kbps)
	G.729ab
	iLBC
	Codec Pre-selection Code
	Voice Processing per SIP Service – TX/RX Audio Gain, Echo Cancellation
	Adjustable Audio Frames per Packet
	Codec Name Assignment
	Codec Profile per SIP SP (2) & OBiTALK Service
	Dynamic Audio Payload
	Packet Loss Concealment
	Jitter Buffer (Adaptive)
	STUN
VoIP Cont…	ICE
	SUBSCRIBE / NOTIFY Framework (RFC 3265)
	NOTIFY Dialog, Line Status
	SUBSCRIBE Message Summary
	VoIP NAT Interworking
	DATE Header Support
	ALERT-INFO Header Support
	Remote-Party-ID (RPID)
	P-Asserted-Identity (PAID)
	RTP Statistics in BYE Message
	Media Loopback Support
	T.38 FAX Relay
	FAX and Modem Passthrough
	Configurable Contact List (Inbound Call Routing)
	Automatic Attendant (English) with Configurable Answer Delay
	PIN Access Control to AA (Up to 4 PINs)
	Recursive Digit Map for Call Routing (AA, Phone, Voice Gateways, Trunk Groups)
	AA Configurable Outbound Call Routing Rule
	SIP Service Configurable Inbound Call Routing Rule (2)
	Direct / Single-Stage Dialing (Route to Voice Gateway)
	Fax Pass Through (G.711)
	Modem Pass Through (G.711)
	In-Band DTMF (G.711)
	Out of Voice Band DTMF (RFC 2833)
	Out of Voice Band DTMF (INFO Method)
	Call Progress Tone Generation
	Tone Profile per SIP SP and OBiTALK service
	Ring Profile per SIP SP and OBiTALK service
	Star Code Profile per SIP SP and OBiTALK service
Telephony	Full Duplex Audio
	G.165, 168 Echo Cancelation
	VAD – Voice Activity Detection
	Silence Suppression
	Comfort Noise Generation
	Three Way Conference Calling with Local Mixing
	Hook Flash Event Signaling
	Flash Hook Timer
	Caller ID – Name & Number per Bellcore, ETSI and DTMF
	MWI – Message Waiting Indicator
	Visual Message Waiting Indication (VMWI)
	Daylight Savings Time Support – North & South Hemispheres
	Caller ID Enable /Disable
	Caller ID Number
	Caller ID Name (Alphanumeric)
	Caller ID Spoofing
	Call Waiting

Network – Application Details – Continued ...	Maximum Session Control Call Forward - Unconditional Call Forward on Busy Call Forward on No Answer (Ring Count Configurable) Call Transfer Enable / Disable Anonymous Call Block Anonymous Call Do Not Disturb Call Return Repeat Dialing
Call Progress Tones	Configurable Call Progress Tone Call Progress Tone Profiles (2) Dial Tone Busy Tone Ringback Tone Reorder Tone Confirmation Tone Holding Tone Second Dial Tone Stutter Tone Howling Tone Prompt Tone Call Forwarded Tone Conference Tone SIT Tones (1-4) Ringing & Call Waiting Tone Configuration Ring Patterns (10) - Configurable Call Waiting Tone Patterns (10) - Configurable Call Waiting Tone Pattern Profiles (2)
Star Code Configuration	Configurable Start Codes Star Code Profiles (2) Redial Call Return Activate Block Caller ID Deactivate Block Caller ID Block Caller ID Once Unblock Caller ID Once Activate Call Forwarding (All Calls) Deactivate Call Forwarding (All Calls) Activate Call Forward on Busy Deactivate Call Forward on Busy Activate Call Forward on No Answer Deactivate Call Forward on No Answer Activate Block Anonymous Calls Deactivate Block Anonymous Calls Activate Call Waiting Deactivate Call Waiting Activate Do Not Disturb Deactivate Do Not Disturb Activate Repeat Dial Deactivate Repeat Dial

Interfaces & Indicator Lights

Internet (WAN)	3 x 10/100BaseT Ethernet Port (802.3) 1 x 10/100/1000BaseT Ethernet Port
Phone (FXS)	8 x RJ-11 FXS Analog Phone Port
Reset Button	Yes
LEDs	14 – Power, Status, 4 x Ethernet Activity, 8 x Phone

LED Indications	Power On, Device Status, Upgrade in Progress Status, Packet RX/TX, Phone Port Status
Audio Line In/Line Out	1 x 3.5mm Audio Line In, 1 x 3.5mm Audio Line Out
USB	3 x USB 2.0 Port (2 in front, 1 in the back)

Certifications

FCC Part 15	
A-Tick	
CE	
ICES-003	
RoHS	Yes
WEEE	Yes
UL/cUL	Yes – Power Adapter

Environmental

Operating Temperature	0º to 45º C (32º to 113º F)
Storage Temperature	-25º to 85º C (-13º to 185º F)
Operating Humidity	10% to 90% Non-condensing
Non-operating Humidity	10% to 90% Non-condensing

Physical Attributes

Dimensions: (width x depth x height)	6.5 x 9.0 x 2.2 centimetres 2.6 x 3.5 x 0.9 inches
Unit Weight:	200 grams / 7 ounces
Shipping Weight	312 grams / 12 ounces (Including Power Supply, Cable and Packaging)
Mounting	Desktop Mountable

Power Supply

Type	Universal Switching with Fixed US, EU, UK or AU Style Plug Prongs (Model Dependent)
Input Power	AC Input: 100 to 240 Volts 0.3A 50-60Hz (26-34 VA)
Output Power	DC: +12V 1.0 Amp Max

Carton Specifications

Units Per Carton	8 Units
Carton Dimensions (width x depth x height)	43.2 x 25.4 x 21.6 centimetres 17 x 10 x 8.5 inches
Carton Weight	6.4 Kilograms / 14 pounds
Cartons Per Std. 20 / 40 ft Container	1,170 / 2,430 Cartons – Non-palletized

Miscellaneous

Requirements	Active Internet Connection Analog Touch Tone Phone Access to Internet Via a Switched Ethernet Port on Home or Office Router (Optional) Active Internet Phone Service Subscription with All Required SIP Credentials to Make & Receive Calls
Documentation	Quick Start / Installation Guide User / Administrative Guide Implementation Guide for Service Providers **
Package Contents	OBi508 Voice Service Bridge and Telephone Adapter Power Adapter 1 x RJ-45 Ethernet Cable (80 inches / 203 centimeters)

	Quick Start / Installation Guide
Warranty	1-Year Hardware (Limited)
Engineering & Design Location	California, USA
Country of Origin	China
HST Code	8517.62.00
Data Sheet State	All content subject to change. This data sheet is not a warranty.
Data Sheet Version	140911.100.1

** For Service Providers Only

iPhone, iPad and iPod touch are trademarks of Apple Computer, Inc.
Google Voice, Google Talk and Android are trademarks of Google, Inc.
OBi, OBiAPP, OBiON and OBiTALK are trademarks of Obihai Technology, Inc.
All other trademarks mentioned in this document are property of their respective owners.
This document is provided by Obihai Technology for planning purposes only. No warranty is implied.

Do Not Use For Emergency Service Calls
Obihai Technology does not warrant the availability or quality of the OBiTALK network. Furthermore, Obihai
Technology will not be liable to you or any third party for any costs or damages arising directly and or indirectly
from the use of this product's hardware & software including without limits any damage, or for any loss whatsoever.