PORTA ONE

PortaSIP

XML / JSON API Reference

65
MAINTENANCE
RELEASE

Copyright Notice & Disclaimers

Table of Contents

Preface

This document provides information for developers who want to interface their applications with PortaSIP® media applications via XML and JSON API. The PortaBilling® XML and JSON API is described in the **PortaBilling XML / JSON API Reference**.

Where to get the latest version of this guide

The hard copy of this guide is updated upon major releases only, and does not always contain the latest material on enhancements that occur in-between minor releases. The online copy of this guide is always up to date, and integrates the latest changes to the product. You can access the latest copy of this guide at: **www.portaone.com/support/documentation/**.

Conventions

This publication uses the following conventions:

Exclamation mark draws your attention to important actions that must be taken for proper configuration.

NOTE: Notes contain additional information to supplement or accentuate important points in the text.

Trademarks and Copyrights

PortaBilling®, PortaSIP® and PortaSwitch® are registered trademarks of PortaOne, Inc.

What is New in Maintenance Release 65?

- An API user can now execute JSON requests in the dry run mode by passing the *dry_run* flag in the *aux_info* structure. The dry run mode does not execute the method itself. Instead, it checks input arguments according to the schema validation rules and returns validation results.
- The **MenuInfo** structure has been extended with the *replay_menu_times, first_digit_timeout, next_digit_timeout, direct_dial_enabled* attributes. If the *direct_dial_enabled* attribute is enabled, then the *action* attribute in the **TransitionInfo** structure cannot acquire the "DirectDial" value.
- The *queue_info* attribute has been added to the **SIPCallInfo** structure.

1. XML / JSON API Overview

Security

Connection to the XML / JSON API interface is provided via HTTPS. Authentication is done using authentication pairs (login-password). Each request to the API should contain the **auth_info** structure as the header attribute. Note that we strongly recommend using the *session_id* property (which must be received during the authorization via the **LoginRequest**) in the **auth_info** structure for all session requests. Otherwise, if you use login-password authentication pairs for every request, new sessions will be created and cause additional load to the database.

XML API

XML (SOAP) API has its own advantages and drawbacks as compared with JSON API. Among the benefits are the following:

- There is a wide range of reusable software available to programmers to handle XML so they do not have to re-invent code.
- XML (SOAP) is more verbose compared with JSON, but because of this, the data encoding result is typically larger than the equivalent encoding in JSON API.

Access to XML API

Proxy (URL): **https://web-server.yourdomain.com:8443/soap/soap.fcgi**
SOAP URI (namespace): **https://web-server.yourdomain.com/UM/SOAP/**

Please replace the **web-server.yourdomain.com** with the actual hostname of your PortaSwitch® web server.

Error Handling

SOAP faults are used to carry error information within a SOAP message. If the actual response has a SOAP fault element as the body entry, then an error has occurred. In this case, the accuracy of any other fields in the response cannot be guaranteed, and you should only use the fault sub-elements to identify the error. Currently, these sub-elements are as follows:

- **faultcode** is intended for use by the client software and provides an algorithmic mechanism for identifying a fault.
- **faultstring** provides a human-readable explanation of a fault, and is not intended for algorithmic processing.

JSON API

As an alternative to XML API, PortaSwitch® supports JSON API, thus providing your development department with a choice of Web Application Services that can be used. Among the advantages of JSON API are the following:

- Simple data structures that can be easily read and written.
- JSON format is faster in parsing and generating data due to simple syntax, thus there is little influence on web server performance.
- Supports the same methods as those in the SOAP.
- Simplifies the creation of front-end web sites that receive and modify data with minimum impact on performance.

Access to JSON API

All JSON requests to PortaSIP® Media Server API must be sent to the following URL: **https://<web-server.yourdomain.com>:8443/rest/<service>/<method>/**

Please replace the **web-server.yourdomain.com** with the actual hostname of your PortaSwitch® web server.

Replace <**service**> with the API service that contains the required method (e.g. specify the **SMPreferences** service to manage voice mailbox preferences.)

Replace <method> with the required API method (e.g. specify **set_folder_preferences** method in order to change mailbox folder preferences.)

Here is an example of the URL the POST request to be sent to:

https://demo.portaone.com:8443/rest/SMPreferences/set_folder_preferences/

Please note that values that have int or float types (according to this guide), will be represented in the JSON responses by a string containing a number, like in the following example:

```
{ "i_menu": "12" }
```

You can run JSON requests in the dry run mode. The dry run mode does not execute the method itself. Instead, it checks input arguments according to the schema validation rules and returns validation results. To run a request in the dry run mode, add the `aux_info` structure into the request. The structure has the following fields:

- `dry_run` – indicates that the method must be run in the dry run mode.

Sending an HTTP request

For HTTP requests you must include the following parameters (in JSON format) in the POST request body:

- `auth_info` – The mandatory authentication information (see the **Security** section).
- `params` – A set of method parameters (in JSON format) that depend on a method structure. Note that method parameters and their structures are the same as those in the SOAP.

Please note that special characters in a URL must be escaped.
For example, if you want to send the request:

```
https://111.111.11.11:8443/rest/AutoAttendant/set_menu_transiti
on/{"login":"000111222","password":"mysEcReTp@ss","session_id":
null,"domain":"111.111.11.11"}/{"i_menu":12,"transition_info":"
event":"#","action":"Transfer","target_i_menu":0,"destination":
"1","play_prompt":"Y"}}
```

it must be transformed into the following form:

```
https://111.111.11.11:8443/rest/AutoAttendant/set_menu_transiti
on/%7B%22login%22%3A%22000111222%22,%22password%22%3A%22
mysEcReTp@ss%22,%22session_id%22%3Anull,%22domain%22%3A%22111.1
11.11.11%22%7D/%7B%22i_menu%22%3A12%22transition_info%22%3A%7B%
22event%22%3A%22%2523%22%2C%22action%22%3A%22Transfer%22%2C%22t
arget_i_menu%22%3A0%2C%22destination%22%3A%221%22%2C%22play_pro
mpt%22%3A%22Y%22%7D%7D
```

The Content-Type header field used with a HTP POST request must have one of the following values:

- application/x-www-form-urlencoded
- multipart/form-data

Error Handling

If the server returns the '500 Internal Server Error' status code in the HTTP response, then the response body contains a JSON object which includes two elements (keys) that carry error information:

- **faultcode**, that is intended for use by the client software and provides an algorithmic mechanism for identifying the fault.
- **faultstring**, that provides a human readable explanation of the fault, and is not intended for algorithmic processing.

WSDL

Each PortaSIP® Media Server has its own set of WSDL documents available for download from the web server. These documents can be downloaded from:

- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Session.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Types.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Voicemail.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=SMPreferences.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=DialDirectory.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=AutoAttendant.xsd**
- **https://web-server.yourdomain.com:8443/soap/wsdl.fcgi?get=Conference.xsd**

All requests to PortaSIP® Media Server API are handled via an SSL connection. By default, PortaSIP® Media Server installations contain a self-signed certificate that provides the means to encrypt data. However, since this certificate's authenticity cannot be validated, you may experience some problems when connecting to an SSL site. In that case, it may be necessary to obtain a certificate from a genuine certificate authority. Another option is to generate your own certificate authority and have certificates deployed to all API clients. However, this goes beyond the scope of the present document.

2. Reference

Notation Conventions

The following typographic conventions apply throughout this chapter:

- * – A value can be entered for this property only when inserting new records and cannot be changed later.
- ** – This property is read-only, and its value cannot be changed.
- Mandatory properties (whose value must be entered during insertion, and cannot be set to an empty value later) are underlined.
- ⁿ – This property can be used with the **nil** attribute to indicate that it is blank (has no content):
    - In the *Request* message the **xsi:nil="true"** attribute can be used to clear the property (set value to NULL in the database).
    - In the *Response* message a property has the **xsi:nil="true"** attribute if it is blank (has the NULL value in the database).

Establishing an Authenticated Session

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Session

Methods

login

Parameters: **LoginRequest**
Return value: **LoginResponse**

Checks the validity of login and password and returns session_id on success. An API fault is generated on failure.

logout

Parameters: **LogoutRequest**
Return value: **LogoutResponse**

Terminates the session. You should call logout() to terminate the session properly.

Type Reference

LoginRequest structure

Property	Type	Description
login	string, 32 char max	Account ID specified on web interface
domain	string	Media Server Domain corresponding to billing environment that the account belongs to
password	string, 16 chars max	Password specified on web interface

LoginResponse structure

Property	Type	Description
session_id	string, 32 chars max	ID of newly opened session

LogoutRequest structure

Property	Type	Description
-	-	-

LogoutResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

Global Methods and Types

Type Reference

The structure below is used to pass authentication data to the API method. There are two possible ways to authenticate an API request: create a session and pass session_id in auth_info, or pass all the required credentials (login/domain/password) in every API request.

auth_info structure

Property	Type	Description
login	string, 32 chars max	Account ID specified on web interface

domain	string	Media Server Domain corresponding to current billing environment
password	string, 16 chars max	Password specified on web interface
or alternatively:		
session_id	string, 32 chars max	The unique ID of previously opened API session

Access to Voicemail Settings

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Voicemail

Methods

get_vm_settings

Parameters: **GetVMSettingsRequest**
Return value: **GetVMSettingsResponse**

This method enables an API user (account) to get general voicemail settings from the PortaSIP® Media Server database.

set_vm_settings

Parameters: **SetVMSettingsRequest**
Return value: **SetVMSettingsResponse**

This method enables an API user (account) to set general voicemail settings in the PortaSIP® Media Server database.

get_vm_greeting

Parameters: **GetVMGreetingRequest**
Return value: **GetVMGreetingResponse**

This method enables an API user (account) to get the sound prompt for a specified greeting from the PortaSIP® Media Server database. The sound file is returned in a MIME attachment.

set_vm_greeting

Parameters: **SetVMGreetingRequest**
Return value: **SetVMGreetingResponse**

15

This method enables an API user (account) to set the sound prompt for a specified greeting type. The sound file is sent in a MIME attachment.

Type Reference

GetVMSettingsRequest structure

This method doesn't have any parameters.

GetVMSettingsResponse structure

Property	Type	Description
vm_settings	VMSettings	Complete information about general voicemail settings

SetVMSettingsRequest structure

May include **any** of the following properties:

Property	Type	Description
vm_settings	VMSettings	Complete information about general voicemail settings

SetVMSettingsResponse structure

Property	Type	Description
vm_settings_saved	int	1 in case of success

VMSettings structure

Property	Type	Description
password	string	Password for accessing voicemail via IVR
password_ask	string	- yes – ask for password when accessing voicemail via IVR; - no – don't ask for password when accessing voicemail via IVR
prompt_levels	string	PortaSIP® Media Server offers three voice prompt settings in each supported language: - standard - extended - rapid
announce_dt	string	Announce the date and time

		when each voicemail was sent. Values: • yes • no
auto_play	string	Auto-play new voicemail(s) when a call to voicemail is established. Values: • yes • no
greetings	string	Type of greeting for users wishing to leave a voicemail. Values: • standard • extended • personal; • name
fax_file	string	Format for received faxes: • multi_png • multi_tiff • pdf • tiff
ext_email	string, max 128 chars	External email for forwarding, copying, and notifying
ext_email_action	string	Action for external email: • none • forward • notify • copy • fwd_as_attachment
ext_email_vm_fmt	string	Voice message audio format: • au • mp3 (default) • wav
enable_disa	string (Yes/No)	Enable DISA functionality for customer's voicemail
disa_password	string	Password for using DISA functionality

GetVMGreetingRequest structure

Property	Type	Description
greeting_type	string	Values: • standard • extended • personal

		• name

GetVMGreetingResponse structure

Property	Type	Description
filename	string	Filename of greeting attached to SOAP response in a MIME attachment

SetVMGreetingRequest structure

Property	Type	Description
greeting_info	GreetingInfo structure	Complete information about general greeting's settings

GreetingInfo structure

Property	Type	Description
greeting_type	string	Values: • extended • personal • name
filename	string	Filename of greeting attached to SOAP request in a MIME attachment

SetVMGreetingResponse structure

Property	Type	Description
success	int	1 in case of success

Access to Folder Preferences and MailBox and Message Display Options

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/SMPreferences

Methods

get_folder_preferences

Parameters: **GetFolderPreferencesRequest**
Return value: **GetFolderPreferencesResponse**

This method enables an API user (account) to get the preferences of his mailbox.

set_folder_preferences

Parameters: **SetFolderPreferencesRequest**
Return value: **SetFolderPreferencesResponse**

This method enables an API user (account) to set the preferences of his mailbox.

get_display_preferences

Parameters: **GetDisplayPreferencesRequest**
Return value: **GetDisplayPreferencesResponse**

This method enables an API user (account) to get the display preferences of his mailbox and messages.

set_display_preferences

Parameters: **SetDisplayPreferencesRequest**
Return value: **SetDisplayPreferencesResponse**

This method enables an API user (account) to set the display preferences of his mailbox and messages.

GetFolderPreferencesResponse structure

Property	Type	Description
folder_prefs	**FolderPreferences** structure	Complete information about the folder preferences; for more information, see below

FolderPreferences structure

Property	Type	Description
trash_folder	string	An IMAP folder where messages are moved on deletion. The messages are deleted completely if this field is set to "none"
draft_folder	string	An IMAP folder where the user can save a message in progress as a draft. The messages aren't saved if this field is set to "none"
unseen_type	int	The Unread Message Notification Type: • 1 – Only Unseen • 2 – Unseen and Total
unseen_notify	int	Enable Unread Message Notification: • 1 – No Notification • 2 – Only INBOX • 3 – All Folders

sent_folder	string	An IMAP folder messages are copied to after they are sent. The messages aren't copied if this field is set to "none"
unseen_cumulative	int	Enable the Cumulative Unread Message Notification. This controls the behavior of the message counter displayed next to each folder in the folder list. When enabled, if the folder contains sub-folders and is collapsed, then the message count includes all messages within all the sub-folders of that folder.
search_memory	int	Memory Search options. If the user searches the mailbox, the search can be saved for quick access later on. This option defines how many mailbox searches will be saved.

Allowed Values: 0 (disabled), 1, 2, 3, 4, 5, 6, 7, 8, 9 |

SetFolderPreferencesRequest structure

Property	Type	Description
folder_prefs	**FolderPreferences** structure	Complete information about the folder preferences; for more information, see below

SetFolderPreferencesResponse structure

Property	Type	Description
success	int	1 in case of success

GetDisplayPreferencesRequest structure

GetDisplayPreferencesResponse structure

Property	Type	Description
display_prefs	**DisplayPreferences** structure	Complete information about the display preferences; for more information, see below

DisplayPreferences structure

Property	Type	Description
wrap_at	int	Defines how many characters to allow before wrapping text
truncate_sender	int	Specifies the lenght of the From / To fields (0 for full)

show_xmailer_default	int	When viewing a message, this displays which email service or client the sender used	
editor_height	int	Specifies the height of the Editor Window	
mdn_user_support	int	Specifies whether to enable the Mail Delivery Notification	
truncate_subject	int	Specifies the lenght of the Subject Field (0 for full)	
body_quote	string	Prefix each line of the original message with this symbol when replying or forwarding an email message	
include_self_reply_all	int	Specifies whether to include user's address in CC when he chooses Reply All	
sig_first	int	Specifies whether to prepend signature before Reply/Forward text	
pf_cleandisplay	int	Specifies whether to display the View Printable Version link in a message	
editor_size	int	Specifies the width of the Editor Window.	
show_html_default	int	Specifies what version to show by default if a received message is sent in both text and HTML formats: 0 – Text version 1 – HTML version	
page_selector_max	int	Specifies the maximum number of pages that will be shown at one time	
internal_date_sort	int	Specifies whether to sort messages by Received Date	
page_selector	int	Specifies whether to show Page Selector. When enabled, message pages will be shown above and below the list of messages, allowing the user to quickly jump to a specific message page	
addrsrch_fullname	string	Specifies the format of addresses added from the address book: • "Noprefix" – No prefix, address only • "Nickname" – Nickname and address • "Fullaname" – Full name and address	
show_num	int	Specifies the number of messages that will be shown on one page	
show_images	int	Specifies whether to display attached	

		images with the message

SetDisplayPreferencesRequest structure

Property	Type	Description
display_prefs	**DisplayPreferences** structure	Complete information about the display preferences; for more information, see below

SetDisplayPreferencesResponse structure

Property	Type	Description
success	int	1 in case of success

Auto Attendant Configuration

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/AutoAttendant

Methods

get_menu_list

Parameters: **GetMenuListRequest**
Return value: **GetMenuListResponse**

This method enables an API user (account) to get a list of all configured auto attendant menus.

update_menu

Parameters: **UpdateMenuRequest**
Return value: **UpdateMenuResponse**

This method enables an API user (account) to update the settings of a separate auto attendant menu.

create_menu

Parameters: **CreateMenuRequest**
Return value: **CreateMenuResponse**

This method enables an API user (account) to create an auto attendant menu.

del_menu

Parameters: **DelMenuRequest**

Return value: **DelMenuResponse**

This method enables an API user (account) to delete an auto attendant menu.

set_menu_prompt

Parameters: **SetMenuPromptRequest**
Return value: **SetMenuPromptResponse**

This method enables an API user (account) to set (record) separate promt for selected auto attendant menu. The sound file is sent in a MIME attachment to the API request.

get_menu_prompt

Parameters: **GetMenuPromptRequest**
Return value: **GetMenuPromptResponse**

This method enables an API user (account) to get a separate prompt from the selected auto attendant menu. The sound file is sent in a MIME attachment to the API request.

get_menu_transition_list

Parameters: **GetMenuTransitionListRequest**
Return value: **GetMenuTransitionListResponse**

This method enables an API user (account) to get a list of auto attendant menu transitions.

set_menu_transition

Parameters: **SetMenuTransitionRequest**
Return value: **SetMenuTransitionResponse**

This method enables an API user (account) to set auto attendant menu transitions. The transition prompt should be sent in a MIME attachment.

get_menu_transition_prompt

Parameters: **GetMenuTransitionPromptRequest**
Return value: **GetMenuTransitionPromptResponse**

This method enables an API user (account) to get an auto attendant menu transition prompt. The prompt is sent in a MIME attachment.

set_menu_transition_promt

Parameters: **SetMenuTransitionPromptRequest**

Return value: **SetMenuTransitionPromptResponse**

This method enables an API user to set an auto attendant menu transition prompt. The transition prompt should be sent in a MIME attachment.

Type Reference

GetMenuListRequest structure

Property	Type	Description
-	-	-

GetMenuListResponse structure

Property	Type	Description
menu_list	array of **MenuInfo** structures	The list of auto attendant menus

UpdateMenuRequest structure

Property	Type	Description
menu_info	**MenuInfo**	Auto attendant menu data

UpdateMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record

CreateMenuRequest structure

Property	Type	Description
menu_info	**MenuInfo**	Auto attendant menu data

CreateMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of created menu record

DelMenuRequest structure

Property	Type	Description
i_menu	int	The unique ID of deleted menu record

DelMenuResponse structure

Property	Type	Description
i_menu	int	The unique ID of deleted menu database

		record

MenuInfo structure

Property	Type	Description
i_menu*	int	The unique ID of menu record (required for the update_menu and del_menu methods)
name	string, max 64 chars	**The unique** within one account menu name; 'ROOT' name is reserved for the root menu, which always exists
period	string, max 255 chars	Period in special format (see the **How to Define a Time Period** section of this guide).
period_desc	string, max 255 chars	Description of period in a form understandable by end-users
msg_disabled_type	string	'Unavailable' prompt type – standard or recorded by user. Values: • standard • custom
msg_timeout_type	string	'Timeout' prompt type – standard or recorded by user. Values: • standard • custom
msg_intro_set	int	1 if 'Into' prompt recorded; otherwise 0
msg_menu_set	int	1 if 'Menu' prompt recorded; otherwise 0
msg_disabled_set	int	1 if 'Unavailable' prompt recorded; otherwise 0
msg_timeout_set	int	1 if 'Timeout' prompt recorded; otherwise 0
msg_intro_type	string	'Intro' prompt type – standard or recorded by user. Values: • standard • custom
msg_menu_type	string	'Menu' prompt type – standard or recorded by user. Values: • standard • custom

replay_menu_times	int	The number of times to repaly the menu prompts
first_digit_timeout n	int	The timeout in seconds to wait while the first digit is entered
next_digit_timeout	int	The maximum timeout in seconds between collected digits. Default: 5
direct_dial_enabled	string (Y/N)	If set to Y, allow dialing extension from the menu directly. If enabled, the "DirectDial" value for the *action* attribute will be forbidden. Default: N

SetMenuPromptRequest structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record
prompt_type	string	Prompt type: • intro • menu • disabled • timeout
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuPromptResponse structure

Property	Type	Description
i_menu	int	The unique ID of updated menu record

GetMenuPromptRequest structure

Property	Type	Description
i_menu	int	The unique ID of menu record
prompt_type	string	Prompt type: • intro • menu • disabled • timeout

GetMenuPromptResponse structure

Property	Type	Description
prompt	string	Filename of a prompt that is being sent in a

		MIME attachment to the API response

GetMenuTransitionListRequest structure

Property	Type	Description
i_menu	int	The unique ID of menu record

GetMenuTransitionListResponse structure

Property	Type	Description
transition_list	array of **TransitionInfo** structures	Set of transitions for specified auto attendant menu

SetMenuTransitionRequest structure

Property	Type	Description
i_menu	int	The unique ID of the menu record
transition_info	**TransitionInfo**	Properties of the menu transition

SetMenuTransitionResponse structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record

GetMenuTransitionPromptRequest structure

Property	Type	Description
event	string	Transition event; see allowed values in **TransitionInfo** structure
i_menu	int	The unique ID of menu record
i_menu_transition	int	The unique ID of the menu transition record

GetMenuTransitionPromptResponse structure

Property	Type	Description
prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuTransitionPromptRequest structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record

prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API request

SetMenuTransitionPromptResponse structure

Property	Type	Description
i_menu_transition	int	The unique ID of the menu transition record

TransitionInfo structure

Property	Type	Description
action	string	Performed action. Possible values: • Disabled – No action. • Directory – Launch the 'Dial Directory' IVR. • Queue – Transfer to the call queue specified in the *target_i_menu* property. • Transfer – Transfer to the preconfigured number specified in the *destination* property. • TransferE164 – Transfer to the E164 number specified in the *destination* property. • Voicemail – Launch voicemail recording. • Menu – Go to the auto attendant menu specified in *target_i_menu* property. • Extension – Transfer to the extension dialed by a user. • DISA – Make a call. • DirectDial.

announce_ext_numbers	string	Specifies whether to announce the external number. Possible values: • Y – Announce the external number. • N – Don not announce the external number.
destination	string, max. 32 chars	Destination for 'Transfer,' 'TransferE164' action
event	string	Transition event. Possible values: '0', '1', '2', '3', '4', '5', '6', '7', '8', '9', '*', '#', 'Timeout', 'Not Active', 'f'.
i_menu_transition	int	The unique ID of the menu transition record
max_size	int	The maximum allowed number of digits that a user can input as an extension (applicable only for the Extension *action*)
play_prompt	string	Play or do not play user-recorded prompt before action. Possible values: • Y • N
prompt	string	Filename of a user-recorded prompt that is being sent in a MIME attachment (only for the set_menu_transition method)
prompt_set	int	1 if user-recorded prompt set
target_i_menu	int	The unique ID of the auto attendant menu record
target_i_queue	int	The unique ID of the call queue to which the call must be transferred.

Conference Configuration

SOAP URI: https://web-server.yourdomain.com/UM/SOAP/Conference

Methods

get_conf_info

Parameters: **GetConfInfoRequest**
Return value: **GetConfInfoResponse**
Realm: account

This method enables an API user to obtain conference settings by i_conf or name.

get_conf_list

Parameters: **GetConfListRequest**
Return value: **GetConfListResponse**
Realm: account

This method enables an API user to obtain a list of all his conferences and their settings.

create_conf

Parameters: **CreateConfRequest**
Return value: **CreateConfResponse**
Realm: account

This method enables an API user to create a new conference entity.

update_conf

Parameters: **UpdateConfRequest**
Return value: **UpdateConfResponse**
Realm: account

This method enables an API user (account) to update a conference entity.

del_conf

Parameters: **DelConfRequest**
Return value: **DelConfResponse**
Realm: account

This method enables an API user to delete a certain conference.

set_conf_prompt

Parameters: **SetConfPromptRequest**
Return value: **SetConfPromptResponse**

This method enables an API user (account) to set (record) separate prompts for conferences. The sound file is sent in a MIME attachment to the API request.

get_conf_prompt

Parameters: **GetConfPromptRequest**
Return value: **GetConfPromptResponse**

This method enables an API user (account) to get a prompt recorded for a conference. The sound file is sent in a MIME attachment to the API request.

Type Reference

ConfInfo structure

Property	Type	Description
i_conf	int	The unique ID for a conference entity
name	string	A conference name
pin_host	string	PIN for administrator to log into the conference
pin_user	string	PIN for user to join the conference
max_call_duration	int	Maximum conference duration
max_session_time	int	Maximum session time (exclusive with start / end time)
max_participants	int	Maximal participants
wait_host	string	Whether the administrator should log in first. Allowed values: Y, N
play_announce	string	Specifies whether announcements should be played. Allowed values: Y, N
play_moh	string	Specifies whether MOH should be played. Allowed values: Y, N
start_time	string	Specifies when the conference will start (For permanent conference use max_session_time). Note that the time is defined in UTC

31

msg_intro_set	string	1 if an 'Intro' prompt recorded; otherwise 0
moh_set	string	1 if a 'MOH' prompt recorded; otherwise 0
video_conf	string	Specifies whether video conference is enabled. Allowed values: Y, N

GetConfInfoRequest structure

Property	Type	Description
i_conf	int	The unique ID for the conference
name	string	The conference name

GetConfInfoResponse structure

Property	Type	Description
conf_info	**ConfInfo** structure	General conference settings

GetConfListRequest structure

Property	Type	Description
-	-	-

GetConfListResponse tructure

Property	Type	Description
conf_list	array of **ConfInfo** structures	The list of conferences and their settings

CreateConfRequest structure

Property	Type	Description
conf_info	**ConfInfo** structure	General conference settings

CreateConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for a new conference

UpdateConfRequest structure

Property	Type	Description
conf_info	**ConfInfo** structure	General conference settings

UpdateConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for the updated conference

DelConfRequest structure

Property	Type	Description
i_conf	int	The unique ID for the conference to be deleted

DelConfResponse structure

Property	Type	Description
i_conf	int	The unique ID for deleted conference

SetConfPromptRequest structure

Property	Type	Description
i_conf	int	The unique ID for a conference record
prompt_type	string	Prompt type: • intro • moh
prompt	string	Filename for a prompt that is being sent in a MIME attachment to the API request

SetConfPromptResponse structure

Property	Type	Description
i_conf	int	The unique ID for the updated conference record

GetConfPromptRequest structure

Property	Type	Description
i_conf	int	The unique ID for a conference record
prompt_type	string	Prompt type: • intro • moh

GetConfPromptResponse structure

Property	Type	Description

prompt	string	Filename of a prompt that is being sent in a MIME attachment to the API response

3. Call Control API

Overview

The Call control API permits to originate, answer, terminate a call, retrieve a list of currently established calls and subscribe to notifications about call state changes for certain customers and accounts. Together with already existing API methods (e.g. for retrieving customer information), these help build a full-grown CTI solution.

Call control API is asynchronous and is accessible via WebSockets. A WebSocket-secure (WSS) connection is established on the web server and the JSON-RPC is used to send requests and receive responses. (Common XML (SOAP) and JSON API can be used as well. However, it does not allow to subscribe to real-time call state changes.)

WebSocket connections are processed by workers. Each worker can process up to 100 concurrent connections. The actual maximum number of connections possible, however, depends upon the capacity and general configuration of the Apache server.

Another feature of a WebSocket connection is its inactivity timeout – the period after which the connection automatically closes. To support a Websocket connection, define some value, e.g., 300 seconds, for the [API]WebSocketInactivityTimeout option on the Configuration server, and make sure that your application can call the **ping** method to renew the session.

The internal communication between the web server and PortaSIP® is performed via the HTTP and Redis protocols, therefore, for real-time notifications of call state changes, the Redis instance must be configured on the Configuration server.

By default, only administrators and reseller users have access to the call control API. To access the API from retail customer and account realms, create a new ACL for them and then define the permissions:

```
Update / Allow / Call_control / *
Read / Allow / Call_control / *
```

Access to JSON-RPC API

All JSON-RPC requests to the API must be sent to the following URL:
wss://<web-server.yourdomain.com>:<port>/ws

⚠️ Please replace the **web-server.yourdomain.com** with the actual hostname of your web server.

Replace **<port>** with the required port. The JSON-RPC interface is available for administrators on port 443, the interface for customers is

available on port 8444, the interface for resellers is available on port 8442 and the interface for accounts is available on port 8445.

Here is an example of the URL the POST request to be sent to:

```
wss://demo.portaone.com:8443/ws
```

Please note that values that have int or float types (according to this guide), will be represented in the JSON responses by a string containing a number, like in the following example:

```
{ "i_menu": "12" }
```

Sending a JSON-RPC request

For JSON-RPC requests you must include the following parameters in the POST request body:
- `cseq` – Since the WebSocket protocol is asynchronous, this value is used to match the response with the request (the same value is present in the response). If no value is passed in the request, no response is expected and none will be returned.
- `auth_info` – The mandatory authentication information (see the **Security** section).
- `service` – The API service that contains the required method.
- `method` – The name of the required API method.
- `params` – A set of method parameters (in JSON format) that depend on a method structure.

Error Handling

In case a request could not be executed or had errors in its structure, the response contains the following error information:
- **code**, that is intended for use by the client software and provides an algorithmic mechanism for identifying the fault.
- **message**, that provides a human readable explanation of the fault, and is not intended for algorithmic processing.
- **details,** that complement the **message** and contain the erroneous object.

WSDL

Each installation of PortaBilling® contains its own set of WSDL documents available for download from the web server from the following URL:
https://web-server.yourdomain.com/wsdl/

Please replace the **web-server.yourdomain.com** with the actual hostname of your PortaSwitch® web server.

Access to Customer Information

SOAP URI (namespace): **https:// web-server. yourdomain.com /Porta/SOAP/Customer**

Methods

enable_api_notifications

Parameters: **EnableApiNotificationsCustomerRequest**
Return value: **EnableApiNotificationsCustomerResponse**
Realm: administrator, reseller, retail customer

This method enables an API user (operator) to subscribe to API notifications for an existing IP Centrex environment.

disable_api_notifications

Parameters: **DisableApiNotificationsCustomerRequest**
Return value: **DisableApiNotificationsCustomerResponse**
Realm: administrator, reseller, retail customer

This method enables an API user (operator) to unsubscribe from API notifications for a specified IP Centrex environment.

Type Reference

EnableApiNotificationsCustomerRequest structure

Property	Type	Description
event	string	The event name. Possible values: sip.call_control_notifications
i_customer	unsignedLong	The unique ID of the customer record
i_main_office	unsignedLong	The unique ID of the customer record that is defined as the main office in the customer hierarchy

EnableApiNotificationsCustomerResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

DisableApiNotificationsCustomerRequest structure

Property	Type	Description

event	string	The event name. Possible values: • sip.call_control_notificati ons
i_customer	unsignedLong	The unique ID of the customer record
i_main_office	unsignedLong	The unique ID of the customer record that is defined as the main office in the customer hierarchy

DisableApiNotificationsCustomerResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

Access to Account Information

SOAP URI (namespace): **https:// web-server. yourdomain.com /Porta/SOAP/Account**

Methods

enable_api_notifications

Parameters: **EnableApiNotificationsAccountRequest**
Return value: **EnableApiNotificationsAccountResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to subscribe to API notifications for a phone line.

disable_api_notifications

Parameters: **DisableApiNotificationsAccountRequest**
Return value: **DisableApiNotificationsAccountResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to disable API notifications for a phone line.

Type Reference

EnableApiNotificationsAccountRequest structure

Property	Type	Description
i_account	unsignedLong	The unique ID of the account or

		account alias record
event	string	The event name. Possible values: • sip.call_control_notifications

EnableApiNotificationsAccountResponse structure

Property	Type	Description
success	int	1 for success, 0 for failure

DisableApiNotificationsAccountRequest structure

Property	Type	Description
i_account	unsignedLong	The unique ID of the account or account alias record
event	string	The event name. Possible values: • sip.call_control_notifications

DisableApiNotificationsAccountResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

Access to Call Information

These methods are accessible both via WebSockets and XML (SOAP) and JSON API.

SOAP URI (namespace): **https:// web-server. yourdomain.com /Porta/SOAP/CallControl**

Methods

get_sip_calls_list

Parameters: **GetSipCallsListRequest**
Return value: **GetSipCallsListResponse**
Realm: administrator, reseller, retail customer, account
Standalone mode support: Yes

This method enables an API user (operator) to receive information about calls made by a particular phone line or by all phone lines within an IP Centrex environment.

originate_call

Parameters: **OriginateCallRequest**
Return value: **OriginateCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to initiate a call for a phone line.

answer_call

Parameters: **AnswerCallRequest**
Return value: **AnswerCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to answer an incoming call.

terminate_call

Parameters: **TerminateCallRequest**
Return value: **TerminateCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to terminate a call.

hold_call

Parameters: **HoldCallRequest**
Return value: **HoldCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to put a call on hold.

unhold_call

Parameters: **UnholdCallRequest**
Return value: **UnholdCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to release a call from hold.

transfer_call

Parameters: **TransferCallRequest**
Return value: **TransferCallResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to transfer a call.

join_calls

Parameters: **JoinCallsRequest**
Return value: **JoinCallsResponse**
Realm: administrator, reseller, retail customer, account

This method enables an API user (operator) to join calls when performing attended transfer of a call.

Type Reference

GetSipCallsListRequest structure

Property	Type	Description
i_account	unsignedLong	The unique ID of the account record
i_main_office	unsignedLong	The unique ID of a customer record that is defined as the main office in the customer hierarchy
i_customer	unsignedLong	The unique ID of the customer record

GetSipCallsListResponse structure

Property	Type	Description
calls_list	Array of **SIPCallInfo** structure	The list of calls

SIPCallInfo structure

Property	Type	Description
call	**SipCallIdentifier** structure	The unique ID of the call
callee	**SipCallerInfo** structure	The information about who the caller is calling to
caller	**SipCallerInfo** structure	The information about who the caller is calling to
connect_time	dateTime	The date and time when the call was connected
duration	int	The call length (in seconds) from the moment when the call was connected
queue_info	**SipCallQueueStateInfo** structure	The information about a call queue status
reason	string	Describes the reason for terminated or not established calls
reason_code	int	The code of the reason the calls

		ended
start_time	dateTime	The date and time when the call was initiated
state	string	Defines the current state of the call
transport_id	string	The SIP transport id
type	string	Defines the type of the call. Possible values: • outgoing • incoming

SipCallerInfo structure

Property	Type	Description
account_id	string	The unique ID of the account
centrex_id	string	The unique ID of a customer record
display_id	string	The display number provided by the callee/caller
display_name	string	The display name provided by the callee/caller
extension_id	string	The account's extension number
huntgroup_id	string	The huntgroup number
id	string	Calling / destination number depending on whether it is an incoming or outgoing call

SipCallIdentifier structure

Property	Type	Description
tag	string	The call remote tag
id	string	The call identifier

OriginateCallRequest structure

Property	Type	Description
i_account	unsignedLong	The unique ID of the account record. Not mandatory if executed under account realm.
cld	string	The destination number
callee_auto_pickup	string (Y/N)	Defines whether to request the auto-answer functionality from the callee's UA

OriginateCallResponse structure

Property	Type	Description
call	SipCallIdentifier structure	The unique ID of the call

success	int	1 if operation is successful, 0 if not

AnswerCallRequest structure

Property	Type	Description
transport_id	string	The SIP address of a call defined in format IP:port
callee_answer_mode	string	The answer method of the callee's device. Possible values: • notify • invite
call	**SipCallIdentifier** structure	The unique ID of the call

AnswerCallResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

TerminateCallRequest structure

Property	Type	Description
transport_id	string	The SIP address of a call defined in format IP:port
call	**SipCallIdentifier** structure	The unique ID of the call

TerminateCallResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

HoldCallRequest structure

Property	Type	Description
call	**SipCallIdentifier** structure	The identifier of the call to be put on hold
transport_id	string	The SIP transport id

HoldCallResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

UnholdCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call to be released from hold
transport_id	string	The SIP transport id

UnholdCallResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

TransferCallRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call to be transferred
cld	string	The transfer target
transport_id	string	The SIP transport id

TransferCallResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

JoinCallsRequest structure

Property	Type	Description
call	SipCallIdentifier structure	The identifier of the call to be joined
to_call	SipCallIdentifier structure	The identifier of the call to be joined with
transport_id	string	The SIP transport id

JoinCallsResponse structure

Property	Type	Description
success	int	1 in case of success, 0 in case of failure

SipCallQueueStateInfo structure

Property	Type	Description
i_c_queue	usingnedLong	The unique ID of a call queue record
position	int	A caller's position in the queue
operators	int	The number of active operators

4. Appendices

Sample Script for PortaSIP® Media Server SOAP Communication

```perl
#!perl -w
use strict;
# to enable client-side script debugging uncomment the line below
# and comment the one next to it
#use SOAP::Lite   +trace => 'debug';

use SOAP::Lite;
use MIME::Entity;
use Data::Dumper;

# If the server certificate is not trusted (e.g. it was not issued by a
# trusted certificate authority), then ignore it.
$ENV{PERL_LWP_SSL_VERIFY_HOSTNAME}=0;

##### Preparing ####################################################

my $soap_sess = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/Session')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_test = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/Voicemail')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_dial_dir = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/DialDirectory')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $soap_aa = SOAP::Lite
    -> uri('https://localhost/UM/SOAP/AutoAttendant')
    -> proxy('https://pum-host:8443/soap.fcgi')
    -> on_fault( sub {
        my($soap, $res) = @_;
        print ("SOAP error:". (ref $res ? $res->faultstring : $soap-
>transport->status . "/" . $res));
        });

my $authInfo = $soap_sess->login({
        'login'     => '88881',
        'domain'    => 'pum.somedomain.com',
        'password'  => 'test123'})->result();
$authInfo = SOAP::Header->name( 'auth_info' => {
                'session_id' => $authInfo->{'session_id'}
                });
my $authInfo_nosess = SOAP::Header->name( 'auth_info' => {
        'login'     => '88881',
        'domain'    => 'pum.somedomain.com',
        'password'  => 'test123'
                });
```

```
####################################################################
my $res;
#example of accessing SOAP module without establishing session
$res = $soap_test->get_vm_settings($authInfo_nosess)->result();
print Dumper($res);
$res = $soap_test->set_vm_settings($authInfo,
        {'vm_settings' =>
            {
                'password'  => '777',
                'ext_email' => 'sergey.pavlov@gmail.com',
                'auto_play' => 'no',
                'announce_dt'=> 'no'
            } })->result();
print "set_vm_settings done\n";
my $ent =  MIME::Entity->build(
                'Filename'      => 'wellcome.au',
                'Type'  => 'audio/basic',
                'Encoding'  => 'base64',
                'Path'          =>
'/var/lib/psmsc/prompts/en/personal_ivr/frw_select_order.au',
                );

my @parts = ($ent);
$res = $soap_test->parts([ $ent ])->set_vm_greeting($authInfo,

        {'greeting_info' =>
            {
                'greeting_type'  => 'name',
                'filename'=> 'wellcome.au'
            } })->result();
print "set_vm_greeting done\n";

$res = $soap_test->get_vm_greeting($authInfo,
            {
                'greeting_type'  => 'name',
            })->result();
print "get_vm_greeting done\n";

######## Dial Directory ###########################

$res = $soap_dial_dir->get_dir_info($authInfo)->result();
print "get_directory_list done\n";
print Dumper($res);

$res = $soap_dial_dir->parts(@parts)->create_dir_entry($authInfo,
    {
    'dir_entry_info'  => {
    'active'              => 'Y',
    'abbreviated_number'=> '1787896',
    'number_to_dial'     => '111111',
    'lastname'            => 'LName',
    'description'         => 'desc foo',
    'prompt'             => 'wellcome.au'
    }
    } )->result();
print "create_dir_entry done\n";
print Dumper($res);
my $i_entry = $res->{'i_entry'};
$res = $soap_dial_dir->parts(@parts)->update_dir_entry($authInfo,
    {
    'dir_entry_info'  => {
    'i_entry'     => $i_entry,
    'active'              => 'Y',
    'abbreviated_number'=> '99',
    'number_to_dial'     => '565656',
    'lastname'            => 'LastName',
    'description'         => 'desc333',
    'prompt'             => 'wellcome.au'
    }
    } )->result();
```

```perl
print "update_dir_entry done\n";
$res = $soap_dial_dir->get_dir_entry($authInfo, {'i_entry' => $i_entry}
)->result();
print "get_dir_entry done\n";
print Dumper($res);

$res = $soap_dial_dir->del_dir_entry($authInfo,
    {'i_entry'  => $res->{'dir_entry_info'}->{'i_entry'}})->result();
print "del_dir_entry done\n";
print Dumper($res);


######## Auto Attendant ##########################

$res = $soap_aa->get_menu_list($authInfo)->result();
if (!$res) {
    print "get_menu_list failed\n";
}
print "get_menu_list done\n";
print Dumper($res);
my $root_i_menu;
foreach my $menu (@{$res->{'menu_list'}}) {
    if ($menu->{'name'} eq 'ROOT') {
        $root_i_menu = $menu->{'i_menu'};
        last;
    }
}

$res = $soap_aa->parts(@parts)->set_menu_prompt($authInfo,
    {
        'i_menu'        => $root_i_menu,
        'prompt_type'   => 'intro',
        'prompt'        => 'wellcome.au'
    })->result();
print "set_menu_prompt done\n";
print Dumper($res);

$res = $soap_aa->set_menu_transition($authInfo,
    {
        'transition_info' => {
        'i_menu'     => $root_i_menu,
        'event'      => '0',
        'action'     => 'Transfer',
        'destination'=> '5555',
        }
    })->result();
print "set_menu_transition done\n";
print Dumper($res);

$res = $soap_aa->get_menu_transition_list($authInfo,
    {
        'i_menu'     => $root_i_menu,
    })->result();
print "get_menu_transition_list done\n";
print Dumper($res);

$res = $soap_aa->get_menu_prompt($authInfo,
    {
        'i_menu'        => $root_i_menu,
        'prompt_type'   => 'intro',
    })->result();
print "get_menu_prompt done\n";
print Dumper($res);

$res = $soap_aa->create_menu($authInfo,
    { 'menu_info' => {
        'name'          => 'AABBBCCC',
        'period'        => 'hr{0-11}',
        'period_desc'   => 'Some period',
        'msg_timeout_type' => 'standard'
        }
    })->result();
print "create_menu done\n";
```

```
print Dumper($res);
my $new_i_menu=$res->{'i_menu'};

$res = $soap_aa->update_menu($authInfo,
    { 'menu_info' => {
        'i_menu'        => $new_i_menu,
        'name'          => 'DDDEEFF',
        'period'        => 'hr{0-2}',
        'period_desc'   => 'New period',
        }
    })->result();
print "update_menu done\n";
print Dumper($res);

$res = $soap_aa->get_menu_list($authInfo)->result();
print "get_menu_list done\n";
print Dumper($res);

$res = $soap_aa->del_menu($authInfo,
    { 'i_menu' => $new_i_menu })->result();
print "del_menu done\n";
print Dumper($res);

$res = $soap_sess->logout($authInfo)->result();
print "logout done\n";
print Dumper($res);
```

How to Define a Time Period

A time period is specified as a string in the following format:

```
sub-period[, sub-period...]
```

A sub-period takes the following form:

```
scale {range [range ...]} [scale {range [range ...]}]
```

The scale must be one of the nine different options (or their equivalent codes):

Scale	Scale Code	Valid Range Values
year	yr	n – where n is an integer 0<=n<=99 or n>=1970
month	mo	1-12 or jan, feb, mar, apr, may, jun, jul, aug, sep, oct, nov, dec
week	wk	1-6
yday	yd	1-365
mday	md	1-31
wday	wd	1-7 or su, mo, tu, we, th, fr, sa
hour	hr	0-23 or 12am 1am-11am 12noon 12pm 1pm-11pm
minute	min	0-59
second	sec	0-59

The same scale type may be specified multiple times. Additional scales simply extend the range defined by previous scales of the same type. The range of a given scale must be a valid value in the form:

v

or

v-v

In the range specification *v-v*, if the second value is larger than the first, the range wraps around unless the scale specification is "year". Year does not wrap because a year is never really reset, rather it just changes by increments.

Ignoring that fact that led to the dreaded Y2K nightmare, when a year rolls over from 99 to 00, it has really rolled over one century, not gone back a century. Time period supports the ambiguous two digit year notation because it is so widespread.

However, two-digit notation is converted to four digits by prepending the first two digits of the current year. In the case of 99-1972, the 99 is translated to whatever the current century is (probably the 20th), and so the range 99-1972 is treated as 1972-1999. For the 21st century, the range would then be 1972-2099.

In any case, if v-v is 9-2, and the scale is month, September, October, November, December, January, and February are the months specified by the range (9-2 is the same as Sep-Feb).

If v-v is 2-9, then the valid months are February, March, April, May, Jun, July, August, and September.

v is not a point in time. For the hour scale, 9 specifies the time period from 9:00:00 am to 9:59:59 am. This is what most people would call 9-10.

In other words, v is discrete in its time scale. 9 changes to 10 when 9:59:59 changes to 10:00:00, but 9 is the period from 9:00:00 to 9:59:59. Just before 9:00:00, v was 8.

Note that there can be a white space anywhere, and case is unimportant. Note also that scales must be specified either in long form (year, month, week, etc.) or in code form (yr, mo, wk, etc.). Scale forms in a period statement may be mixed.

Furthermore, when using letters to specify ranges, only the first two (for weekdays) or the first three (for months) are significant. January is a valid specification for Jan, and Sunday is a valid specification for su. Sun is also valid for su.

Period Examples

Example 1

To specify a time period from Monday through Friday, 9 a.m. to 5 p.m., use the following period:

```
wd {Mon-Fri} hr {9am-4pm}
```

When specifying a range using "-", it is best to think of "-" as meaning "through", i.e. 9 a.m. through 4 p.m., which is the time interval ending just before 5 p.m.

Example 2

To specify a time period from 9 a.m. to 5 p.m. on Monday, Wednesday, and Friday and from 9 a.m. to 3 p.m. on Tuesday and Thursday, use the following period:

```
wd {Mon Wed Fri} hr {9am-4pm}, wd{Tue Thu} hr {9am-2pm}
```

Example 3

To specify a time period that extends from Monday to Friday, 9 a.m. to 5 p.m., but alternates the weeks in a month, use the following period:

```
wk {1 3 5} wd {Mon Wed Fri} hr {9am-4pm}
```

Example 4

For a period that specifies the winter:

```
mo {Nov-Feb}
```

The next example is equivalent to the previous one:

```
mo {Jan-Feb Nov-Dec}
```

as is:

```
mo {jan feb nov dec}
```

or also:

```
mo {Jan Feb}, mo {Nov Dec}
```

and this, too:

```
mo {Jan Feb} mo {Nov Dec}
```

Example 5

To specify a period of every other half-hour, use something like this:
```
minute {0-29}
```

Example 6

To specify the morning, use the following period definition:

```
hour {12am-11am}
```

Please note that '11 a.m.' here is not the 11:00:00 a.m. time point but the 11:00:00 a.m.–11:59:59 a.m. interval.

Example 7

To specify the period that consists of several 5-second blocks:

```
sec {0-4 10-14 20-24 30-34 40-44 50-54}
```

Example 8

To specify every first half-hour on alternating weekdays, and the second half-hour during the rest of the week, use the following period:

```
wd {1 3 5 7} min {0-29}, wd {2 4 6} min {30-59}
```