HipLink® Software

Integration and Programmers Guide

Last Revised June 2013 | Version 4.7

©2013 Semotus, Inc., DBA HipLink Software. All rights reserved

HipLink® Software Integration Guide

HipLink Software Copyright

The *HipLink Integration and Programmers Guide* is for use by licensed Users only. This document contains proprietary information owned by HipLink Software and is protected by copyright law and international treaties. It may not be copied, published or used, in whole or in part, for any purposes other than as expressly authorized by HipLink Software. Any unauthorized copying, distribution or disclosure of information in any manner without the prior written consent of HipLink Software is a violation of copyright laws and will be prosecuted to the maximum extent possible under law.

Trademarks and Copyright

- HipLink® is a registered trademark of Semotus, Inc., DBA HipLink Software.
- Apache® is a registered trademark of Apache Software Foundation.
- ©Minicom: Copyright 1991, 1992, 1993, 1994, 1995, 1996. Miquel van Smoorenburg.
- All other trademarks and copyrights are the property of their respective owners.

Disclaimer

HipLink Software has made every effort to ensure the accuracy of information contained within this document. However, HipLink Software makes no warranties with respect to this document and disclaims any implied warranties of merchantability or fitness for a particular purpose. All screen images used in this document are for illustrative purposes and are only intended to provide an example of the screen. Screens may vary dependent upon the service provided. Information in this document is subject to change without notice.

Please carefully read the instructions provided in this guide before installing and administering HipLink Software. Retain these instructions for future use. If you do not agree with the terms of this license agreement, do not install, copy, or use this software.

For technical assistance, contact HipLink Technical Support:

Phone: 408-399-0001
Email: support@hiplink.com

HipLink Software Offices
718 University Ave, Suite 110
Los Gatos, California 95032
Tel: (408) 399-6120
Fax: (408) 395-5404
Web: www.hiplink.com

Table of Contents

©HipLink Software 2013. All Rights Reserved.

©HipLink Software 2013. All Rights Reserved.

©HipLink Software 2013. All Rights Reserved.

Introduction

Overview

HipLink provides two main categories of interfaces: the main graphical user interface (GUI) and the non-GUI interfaces. These non-GUI interfaces are: a command line interface (CLI), a COM interface and a Java API (available for Windows platforms only), a File System Interface (FSI), the HipAdmin and the SOAP interface.

1. The GUI can be accessed through a web browser (e.g., Internet Explorer) at a local web address and it is used for administration and messaging services.
2. The CLI can be accessed either by a 3rd party application or by other users, locally or remotely.
3. The COM interface, the Java API, and the FSI are used for integration with other applications.
4. The HipAdmin and the SOAP interface can be accessed either by a 3rd party application or by other users, locally or remotely.

This guide describes how to install and use the CLI, the COM interface, the Java API, the FSI, the HipAdmin and SOAP interface. It also presents various Gateways and the Voice XML used by the HipLink Voice Module.

For details about the GUI interface, refer to the HipLink User Guide. The HipLink server installation and administration is detailed in the HipLink Installation and Administration Guide.

HipLink allows setting response action for two-way messages. The action settings chapter in the present guide explains the definition of these action commands and the integration with the HipLink system.

The stand alone HipLink Voice Module allows HipLink to receive incoming calls and to send outgoing voice calls. The Voice XML chapter presents how to program the Voice Module.

Prerequisites

The CLI, COM, Java API, and the FSI non-GUI interfaces are dedicated for sending purposes only. The HipAdmin and SOAP interfaces are dedicated for administration purposes.

All the HipLink entities (i.e., Users, Receivers, Carriers, Messengers, etc.) should be created before attempting to use these interfaces for sending messages.

In order to be able to use the CLI interface, the HipAdmin, or any other non-GUI interface from the HipLink server or from a remote computer, a designated User has to be set up in the HipLink system using the GUI.

The procedure is briefly explained below. For more details about creating Users using the GUI refer to the HipLink User Guide.

There are three types of HipLink Users that can be set only from the GUI: GUI, Non GUI, or both (meaning both GUI and Non GUI). The HipLink License Key allows the administrator to create a limited number of GUI and Non GUI Users. A User of type both will be counted in both categories.

©HipLink Software 2013. All Rights Reserved.

The HipLink administrator has to create one designated User of type Non GUI or both per each remote computer with a unique IP address. The permissions of this User will be the permissions of the User Group to which this User is assigned.

The non-GUI interfaces are not checking the HipLink User identity and password. Every message sent from this IP address will be associated with the designated User assigned to this machine. No distinction is made between messages sent using the CLI, FSI, COM, or Java API interfaces. It is up to the HipLink administrator together with the network administrator to restrict the access to a machine for specific HipLink Users.

For example, the CLI HipAdmin is installed on the Database Server and the HipLink server on the Messaging Server, and a designated User, Database User of type Non GUI was defined using the Database Server IP address. If now the HipLink administrator admin logs in to the database Server and sends a test message using the CLI, in the HipLink system, this test message will be signed Database User and not admit.

If the Database User was defined by mistake using a wrong IP address (i.e., not the Database Server IP address), then the HipLink server will return the following error message: “This IP address does not have send permissions”.

©HipLink Software 2013. All Rights Reserved.

CLI Interface

Introduction

The HipLink CLI (Command Line Interface) is an executable program that sends messages to the HipLink server. CLI is a one-line command that can be executed either by other processes or by a HipLink User. The syntax and return values are described with explanations for each parameter.

In the Examples for Programmers section, there are listed some short examples of shell scripts, Perl scripts, and ANSI C source code that show the utilization of the CLI interface and the handling of its return values.

Installation

The CLI can be used either from the same machine where the HipLink server is installed, or from a different machine.

The CLI comes with the HipLink server installer. Thus, no other installation is required if the CLI is used on the HipLink server. The executable is located in the HipLink/bin directory:

- 'C:\Program Files\HipLink Software\HipLink\bin' on Windows, and
- '/usr/local/hiplink/bin' on Unix platforms.

If the CLI has to be used remotely as a client from a different machine, then it should be copied on that machine. You have to know the URL of the machine where the HipLink server is located.

Note: In order to be able to use the CLI interface, a designated User of type Non GUI or both has to be set up in the HipLink system. This User specifies the IP of the machine from where the CLI will be executed.

For details about configuration refer to the CLI Configuration File section from this chapter.

Parameters

Structure and Syntax

CLI, the Command Line Interface, is an executable program that sends messages to the HipLink server. To specify and define a message the program requires certain parameters that have to be passed to CLI. This section explains the structure and syntax of the CLI and its parameters.

In the table below the CLI parameters are listed in alphabetical order.

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
-action	“action name” | null <mandatory> (after each response option) Defines an action to be executed upon receiving a response linked to it. It must match to the predefined *action* name, defined in the HipLink database (created using GUI). Used for the two-way (SNPP Level 3) messages. If the action name contains spaces, it must be quoted. If there is no action linked to the response, null value must be passed (i.e. –action null). They always come as triplets ‘-response –action –final’ in that specific order.
-attachment	“path” <optional> (to be used only when sending a fax message using the HipLink fax Module, or a voice message using the HipLink Voice Module) Location of the attached file. If the *path* contains spaces, it must be quoted.
-carrier	“carrier name” <mandatory> (only when sending messages with –msgtype quick) Defines the Carrier name for the Quick Message. Must be one of the Carriers defined in the HipLink database (created using GUI). Must be followed by the pin option. If the *carrier name* contains spaces, it must be quoted.
-config_dir	“path” <optional> Location of the configuration file (named cmd_config for all platforms). The User can specify the URL inside this configurations file. If this option (and the URL option) is not used, the default URL as defined in the CLI installation is used. If the *path* contains spaces, it must be quoted.
-confirm	“message ID” <optional> Confirms a message.
-cover	“fax cover page” <optional> (to be used only when sending a fax message using the HipLink Fax Module) Name of the Fax cover page file. If the *fax cover page* file name contains spaces it must be quoted.

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
-email_receipt	“e-mail address” <optional> The reply of a two-way message will be sent to the specified e-mail address.
-final	YES | NO | yes | no <mandatory> (after each response and action option) Defines if a response (linked to an action) is final. Used only for the two-way (SNPP Level 3) messages. If it is set to YES, HipLink server will not continue to query for SNPP Level 3 responses after receiving that response.
-jobid	*jobid* (alphanumeric format) <mandatory> (only if **–msgtype query**) Defines the Job ID for this message. This Job ID overwrites the one generated by the HipLink server and which is guaranteed to be unique. If this feature is used, then the User has the responsibility for the Job ID uniqueness. Used together with **–msgtype query** to check the status of a message.
-log_dir	“path” <optional> Location of the log file. If the path contains spaces, it must be quoted.
-message	“message body” <mandatory> (when sending messages i.e. msgtype different than query). If the message body contains spaces, it must be quoted.
-msgtype	quick | query <optional> Defines the mode of the command. quick- a message will be sent- no Receiver name is required, but 2 additional parameters: -pin and –carrier. query- a query will be sent to the HipLink server to check the message status specified by the –jobid parameter.
-nametime	YES | NO | yes | no <optional> default is yes. Defines if sender name and timestamp should be appended automatically to the message.

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
-param	name=value <optional> (to be used only when sending a voice message with a template using the HipLink Voice Module) Used to pass parameters and their values to a voice template: -param name1=value1 –param name2-value2 If the value of a parameter contains spaces, it must be quoted.
-pin	pin number <mandatory> (only when sending messages with-msgtype quick) Defines the pin number of the Receiver (device). Must follow the Carrier option.
-priority	normal | high <optional>, default is normal Set the priority of the message.
-rcvname	“receiver name” <mandatory> (only if –msgtype is different than quick or query) Defines Receiver name (must match one of the Receivers defined in the HipLink database). If the Receiver name contains spaces, it must be quoted. The –rcvname can be repeated multiple times in the command line.
–response	"*response text*" <optional> (only for two-way messages using SNPP Level 3) Defines a response message. Every response must be followed by the parameters action and final. Multiple response messages can be defined. If the *response text* contains spaces, it must be quoted.
–sender	"*sender name*" <optional> Specifies the name of the sender of the message. This name is used to replace the CLI (i.e., Non GUI) User in the reports. It can be set to any string because the CLI permissions are based on the Non GUI User settings (i.e., IP address of the machine where the CLI is installed). If the *sender name* contains spaces, it must be quoted.
–subject	*“e-mail subject*" <optional> Specifies the subject for a message sent out to Receivers using SMTP protocol. For

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
	Receivers using different protocols the value of the *subject* parameter is ignored. If the *subject* contains spaces, it must be quoted.
–template	"*voice template*" <optional> (to be used only when sending a **voice** message using the HipLink Voice Module) Name of the Voice template file. If the *voice template* file name contains spaces, it must be quoted.
–type	**standard** | **fax** | **voice** <optional> Type of the message. **Standard** – a message to be sent to a wireless device (default value). **Fax** – a voice message to be sent using the HipLink fax Module. **Voice** – a voice message to be sent using the HipLink Voice Module.
–url	*url* <optional> URL of the HipLink server. If this option (and the –**config_dir** option) is not used, the default URL as defined in the CLI installation is used.

Note: All CLI commands are case sensitive.

Note: The maximum length of the command line is platform dependent. It is the User's responsibility to make sure that the length of the CLI command line is not truncated.

Examples

1. Send a simple message to a single Receiver (the path reflects default Unix settings for HipLink):
   -usr/local/hiplink/bin/cli.exe -config_dir /usr/local/hiplink/config -rcvname KIMW –message "Hello world!"
2. Send a simple message to multiple Receivers (the path reflects default Unix settings for HipLink):
   -usr/local/hiplink/bin/cli.exe -config_dir /usr/local/hiplink/config -rcvname KIMW –rcvname "Bill Jones" -message "Hello world!"
3. Send a quick message to a single Receiver, specifying the priority, switching nametime off, and specifying a config_dir (for a different URL):
   path/cli.exe -config_dir /usr/local/hiplink/alternative-config -msgtype quick -priority high - pin 1234567 -carrier SKYTEL1 -nametime NO -message "Hello world!"
4. Send a message to a Group, specifying an alternative URL:
   path/cli.exe -priority high -nametime YES -rcvname ADMINS -message "Hello world!" -url http://123.123.123.123:8000/cgi-bin/action.exe
5. Send a two-way message to a Receiver:

©HipLink Software 2013. All Rights Reserved.

path/cli.exe -config_dir path -rcvname "Bill Jones" -message "Confirm task" -response "I agree" -action ACTION1 -final YES -response "I don't agree" - action ACTION2 -final YES -response "I don't know" -action null -final YES

6. Query for two-way message response (here the -jobid parameter is the return value of the sending in example 5):
   path/cli.exe -config_dir path -msgtype query -jobid 123456_1
7. Send a voice message:
   path/cli.exe -type voice -rcvname admin -message "Service is down"
8. Send a voice message with an attached file:
   path/cli.exe -type voice -rcvname admin -attachment "c:\file.wav"
9. Send a voice message with a template and pass the parameters required by the template:
   path/cli.exe -type voice -rcvname admin -template server_down -param server=f40 -param downtime=10 -param severity=critical
10. Send a voice message with a template and parameters:
    path/cli.exe -type fax -rcvname admin -attachment "c:\resume.doc" -cover cv1

Path

As a general rule, the path should precede the cli.exe command. On UNIX, as well as Windows, the path to the directory, where the cli.exe is installed, can be included into the search-path. If this has been done, the cli.exe can be called without specifying the path. The -config_dir parameter specifies the directory where the cmd_config file is located. If the -url parameter is not used, then the path has to be always specified.

The cli.exe application is automatically installed with the HipLink installer, or can be installed separately, on a different machine, using the HipLink CLI installer.

In both cases, the default path on Unix is: /usr/local/hiplink/bin

On Windows, the default path for the cli.exe installed with the HipLink is: C:\Program Files\Hiplink Software\Hiplink\bin)

Note: The default path for Windows contains white spaces, which can complicate integration into some applications. If you want to avoid this problem it is recommended to install the cli.exe into a different directory, such as C:\HipLink.

CLI Log File Directory

The CLI is keeping a log file. The default location of this log file is that specified in the TEMP variable on both Windows and UNIX platforms. If the TEMP variable is not defined on UNIX, then the "/tmp" directory is used instead. The location of the CLI log file can be changed using the command:

- log_dir parameter.C:\Program Files\HipLink Software\HipLink\bin

Double Quotes Substitution

If there is a need to use double quotes (") in the body of the message, it should be replaced by '%22'. For example the message Say "Hello!" must be written: "Say %22Hello!%22".

©HipLink Software 2013. All Rights Reserved.

Return Values

Status Codes and Descriptions

When the cli.exe is executed, it returns a status code and description, by printing one line of text with 3 or 4 fields separated by white spaces to the standard output.

The return of the CLI has the following format:

<error/status code> <Job_ID> <status text> <possible reply>

Where the errors and status codes are:

No.	Return Code
200	Success
1101	Error – unknown IP
1102	Error – no permission
1103	Error – query failed
1104	Error – Job ID not found
1105	Error – no message created
1106	Error – empty message text
1107	Error – empty receiver
1108	Error – empty group
1109	Error – empty on-duty
1110	Error – empty escalation
1111	Error – empty carrier
1112	Error – empty pin
1113	Error – empty name
1114	Error – empty parameter
1115	Error – invalid action name
1116	Error – no permission
2000	Error – failed to initialize socket
2001	Error – number of predefined responses exceeds limitation
2002	Error – illegal format of response-action-final format
2003	Error – illegal format of final value
2004	Error – invalid message type 11
2005	Error – empty name
2006	Error – empty message
2007	Error – empty job id for query
2008	Error - missing pin
2011	Error – missing URL
2012	Error – HTTP post error

Standard HTTP Protocol OK Codes:

No.	Return Code
100	Continue
101	Switching Protocols
200	OK
201	Created202 Accepted
203	Non-Authoritative Information

©HipLink Software 2013. All Rights Reserved.

204	No Content
205	Reset Content
206	Partial Content
300	Multiple Choices
301	Moved Permanently
302	Moved Temporarily
303	See Other
304	Not Modified
305	Use Proxy

Standard HTTP Protocol Error Codes:

No.	**Return Code**
400	Bad Request
401	Unauthorized
402	Payment Required
403	Forbidden
404	Not Found
405	Method Not Allowed
406	Not Acceptable
407	Proxy Authentication Required
408	Request Time-Out
409	Conflict
410	Gone
411	Length Required
412	Precondition Failed
413	Request Entity Too Large
414	Request-URL Too Large
415	Unsupported Media Type
500	Server Error
501	Not Implemented
502	Bad Gateway
503	Out Of Resources
504	Gateway Time-Out
505	HTTP Version Not Supported

Job ID

HipLink will provide and return a job ID number for each successful message. Upon any error, or if no message could be created, the job ID will be –1.

The job ID will be unique. In case of multiple receivers (for example a group of 3), multiple message files will be created. If the job ID was 123, the message files 123_1, 123_2 and 123_3 will be created. Only the job ID (here: 123) will be returned. A query, however, has to be specified for each individual message (example 123_2). See the examples for details.

Status Text

The CLI can be used to send a query (i.e., -msgtype query) to the HipLink server to check the message status specified by the -jobid parameter.

Example

©HipLink Software 2013. All Rights Reserved.

path/cli.exe -config_dir path -msgtype query -jobid 123456_1

For one-way messages, the status text is New, Failed or Completed.

For two-way messages, the possible statuses are detailed in the following table:

Message	Description
New	Message read by the HipLink messenger and is in the appropriate queue.
Completed	Message successfully delivered to Carrier (for one-way Receivers only).
Pending	Message received by two-way Carrier, but HipLink has not yet received confirmation that it has been delivered to the Receiver.
Delivered	Message successfully delivered to a two-way Receiver.
Replied	Message successfully delivered to two-way Receiver, and HipLink has received a reply (actual reply follows as fourth parameter).
Failed	Send failed.
Confirmed	Message successfully delivered to two-way Receiver, and HipLink has received a confirmation.

Examples

When sending messages:

```
200     55      completed
1107    -1      failed
```

When querying for response (in case of SNPP Level 3):

```
200     55_1    NEW
200     55_1    Delivered
200     55_1    Replied Yes
200     55_1    Replied This text can be longer
1103    55_2    Failed
```

Note: The User can query for every message – not only for SNPP3 messages. In case of other messages, the status will be either Failed or Completed. Completed indicates, that the message has been successfully delivered to the Carrier.

Configuration File

There is a configuration file named cmd_config on all platforms. The file contains the URL of the HipLink Server.

For example: http://123.123.123.123:8000/cgi-bin/action.exe

The User can specify the location of the config file when executing cli.exe.

The -config_dir parameter specifies where config file is located.

For example: -config_dir /usr/local/hiplink/config

©HipLink Software 2013. All Rights Reserved.

Users can also overwrite these settings by alternatively using the **–url** parameter:

-url http://224.224.224.224:8000/cgi-bin/action.exe

Examples for Developers

Using the CLI, the HipLink can be integrated into other 3rd party software applications.

Shell Script Example

```
#!/bin/sh RECE=cindy_email MSG="this is a test."
URL=http://123.123.123.123:8000/cgi-bin/action.exe
CLI=/usr/local/hiplink/bin/cli.exe

S=`$CLI -priority high -nametime YES -rcvname $RECE -message "$MSG" –url $URL`

ERR_CODE=`echo $S|awk '{print $1}'`
JOB_ID=`echo $S|awk '{print $2}'`

if [ $ERR_CODE = 200 ]
then
echo "Message Job_id $JOB_ID Sent Successfully!" else
echo "Message Failed!"
echo "Send failed: $S"
fi
```

Perl Script Example

```
#!/usr/bin/perl

$RECE="cindy_email";
$MSG="this is a test.";
$URL="http://123.123.123.123:8000/cgi-bin/action.exe";
$CLI="/usr/local/hiplink/bin/cli.exe";

my $s=`$CLI -priority high -nametime YES -rcvname $RECE
-message "$MSG" -url $URL`;

my @response = split (/\s+/, $s);

my $err_code = $response[0];
my $job_id = $response[1];
my $descripstion = $response[2];

if ($err_code == 200)
{
print "Message job_id $job_id sent!\n";
}
else
{
print "Message failed!\n";
print "Error Information: $s\n";
}
```

©HipLink Software 2013. All Rights Reserved.

ANSI C Example

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#define URL "http://hiplinkserver:8000/cgi-bin/action.exe" int send_wireless (char *recev, char
*msg)
{
FILE *response; char cmd[1000]; char ret[256];
int err_code;

if ((recev == NULL) || (msg == NULL)) return (-1);

sprintf (cmd, "/usr/local/hiplink/bin/cli.exe "
"–priority high –nametime YES –rcvname %s " "–message \"%s\" –
url %s", recev, msg, URL);

response = popen (cmd, "r");
if (response == NULL) return (–1);

memset (ret, 0, sizeof (ret));
fread (ret, 1, sizeof (ret) – 1, response);
pclose (response); err_code = atoi (ret); return (err_code);
}
```

©HipLink Software 2013. All Rights Reserved.

COM Interface

Introduction

The HipLink COM Interface provides HipLink Users with a Win32 interface for sending wireless messages.

Through this interface, which is similar to the CLI interface, the HipLink product has the ability to receive messages and send them to existing Receivers and/or Groups. The advantage of the COM Interface is that it can be integrated with other customer applications.

The COM object connects to the HipLink server over HTTP (just like the CLI). All parameters (e.g., HipLink IP address or hostname) should be passed as properties of the object. The COM object does not use any config file and does not generate a log file. Error codes and messages are returned by the methods of the COM object.

Installation and Prerequisites

Installation

The HipLink COM Interface can run either on the machine where the HipLink server is already installed (only if it is running on a Windows platform), or on a different Windows machine. HipLink can run with one or more HipLink COM Interfaces as long as they have different IP's.

Download

Download the HipLink COM Interface from the HipLink the Site. Please contact Technical Support for assistance/information with the current download links for the HipLink COM Interface.

Once downloaded, install it on the directory of your choice.

Registration

From the command console, run:

```
regsvr32 HiplinkInterface.dll
```

to register the COM object.

Prerequisites

HIPLINK COM INTERFACE USER

For HipLink, the COM Interface acts like a non-GUI client. For this reason, a non-GUI User account has to be created for the COM Interface (i.e., a User of type Non GUI or both that has an IP address - here the IP address of the machine where the HipLink COM Interface is installed). See the User's Guide for details about creating a User.

Note: Only one Non-GUI User can be assigned to an IP address. This User is allowed to run any Non-GUI HipLink interfaces (i.e., CLI, COM Interface, or E-mail Gateway) from a specific

©HipLink Software 2013. All Rights Reserved.

server. In the Reports, all messages will appear as sent by the same User (i.e., there is no distinction made for the different Non-GUI interfaces).

HiplinkInterface.dll

The HiplinkInterface.dll is a COM object that closely resembles HipLink CLI. Its parameters and usages are virtually identical to CLI. Thus, please refer to the CLI documentation for more details. Typical usage requires setting the various message parameters specified, and then calling Send() function to send the message.

Properties

Name	Description
URL	HipLink host URL.
MsgType	Message type. (see the CLI documentation for details).
ReceName	Receiver name.
TimeStamp	Include time-stamp (“true” or “false”). Default value is “false.”
HighPriority	Has high priority (“true” or “false”). Default value is “false.”
RecePIN	Receiver PIN. Used with *CarrierName* to create a quick message.
CarrierName	Carrier name. Used with RecePin to create a quick message.
Message	Message text.
JobID	Job ID. Used to query the status of a job ID or to overwrite the job ID of a message. It also stores the message job ID if send is successful.

If the COM object used in C++ then Get and Put functions should be used to modify the parameters. The format of this function is: objectName->PutParameterName("parameter value"). e.g., pHiplinkObj->PutRecePIN("123");

If the COM object is used in VB or ASP then the parameters are set directly, like this: objectName.ParameterName = "parameter value". e.g., hipObj.RecePIN = "123"

Methods

Name	Parameters	Return Value	Description
Reset	None	None	Reset all parameters.
GetLastError	None	String	Get last error message string.
GetResult	None	String	Get result string.

©HipLink Software 2013. All Rights Reserved.

Name	Parameters	Return Value	Description
AddRespAction	None	Response text (string) action command (string) final (true/false)	Add response-action pair.
Send	None	True/false	Send the message.

Examples for Developers

The following examples are showing the usage of the HipLink COM Interface from Microsoft C++ and Visual Basic applications.

Examples using Microsoft C++

Example 1

```
IHPCliPtr               pHpCliObj;
_bstr_t                 bstr;
char    buf[ 100 ];

pHpCliObj.CreateInstance("HiplinkInterface.HPCli.1"); pHpCliObj->PutUrl(
"http://Hipserver:8000/cgi-bin/action.exe" );
// create a 2-way message
//

pHpCliObj->PutReceName( "rece-2 way" );
pHpCliObj->AddRespAction( "Yes - I accept", "action 1", FALSE );
pHpCliObj->AddRespAction( "No - I decline", "action 2", TRUE );
pHpCliObj->PutTimeStamp( TRUE );

pHpCliObj->PutHighPriority( TRUE );
pHpCliObj->PutMessage( "Hello! This is a test. Do you accept the test?" );
// send
//

if ( pHpCliObj->Send() == TRUE )
{
// get successful status message
//
bstr = pHpCliObj->GetResult();

wsprintf( buf, "%s", (char *)bstr );
MessageBox( buf, "Ok", MB_OK );
// get job id
//
bstr = pHpCliObj->GetJobID();

wsprintf( buf, "%s", (char *)bstr );
MessageBox( buf, "Job Id", MB_OK );
}
```

©HipLink Software 2013. All Rights Reserved.

```
else
{
// get error message
//
bstr = pHpCliObj->GetLastError();

wsprintf( buf, "%s", (char *)bstr );
MessageBox( buf, "Error", MB_OK );
}

// clean up all properties
//
pHpCliObj->Reset();
```

Example 2

```
// quick message
// pHpCliObj->PutMsgType( "Quick" );

pHpCliObj->PutCarrierName( "Skytel-SNPP" );
pHpCliObj->PutRecePIN( "123" );
```

Example 3

```
// message to a receiver
//
pHpCliObj->PutReceName( "Paul Pager" );
```

Example 4

```
// message to a group
//
pHpCliObj->PutReceName( "support-group" );
```

Example 5

```
// query message status
//
pHpCliObj->PutMsgType( "query" );
pHpCliObj->PutJobID( "HPG370_1.tap" );
```

Examples using Microsoft Visual Basic

Example 1

```
Dim cli As HIPLINKINTERFACELib.HPCli
Dim jobid, msg

'''create a 2-way message

Set cli = New HIPLINKINTERFACELib.HPCli cli.Url =
 "http://Hipserver:8000/cgi-bin/action.exe"
cli.ReceName = "two-way-test-pager"
```

©HipLink Software 2013. All Rights Reserved.

```
cli.AddRespAction "Yes - I accept", "action 1", False
cli.AddRespAction "No - I decline", "action 2", True

cli.Message = "Hello! This is a test. Do you accept the test?"
cli.TimeStamp = True
cli.HighPriority = True if

cli.Send then

'"succeed
jobid = cli.JobID MsgBox
cli.GetResult else
'"failed
MsgBox cli.GetLastError end if

'" clean up
cli.Reset
```

Example 2

```
'"quick message - using PIN and a defined carrier
cli.MsgType = "Quick"
cli.CarrierName = "Skytel-SNPP"
cli.RecePIN = "123"
```

Example 3

```
'"message to a receiver
cli.ReceName = "Paul Pager"
```

Example 4

```
'"message to a group cli.ReceName =
"support-group"
```

Example 5

```
'"query for message status
cli.MsgType = "query"
cli.JobID = "HPG370_1.tap"
```

Example using ASP

```
Dim jobid, msg, cli

Set cli = server.CreateObject("HIPLINKINTERFACE.HPCli.1")
cli.Url = "http://demo.hiplinkwireless.com/cgi-bin/action.exe"
cli.ReceName = "steven"
cli.AddRespAction "Query", "queryevent", 1 ' 1 means final action
cli.Message = "Hello! This is a test. Do you want the list?"
cli.TimeStamp = 1 ' include timestamp
cli.HighPriority = 1 ' set to high priority

if cli.Send = 1 then ' succeed
```

©HipLink Software 2013. All Rights Reserved.

```
        response.write("succeed")
        Response.Write jobid Response.Write cli.GetResult()
else ' fail response.write("failed")
        Response.Write cli.GetLastError()
        end if

        ' clean upcli.Reset
```

HTTP Interface

Introduction

The HipLink HTTP Interface provides HipLink Users with the ability of sending wireless messages using HTTP Post or Get requests from a 3rd party application.

The advantage of the HTTP Interface is that it can be easily integrated with other customer applications and does not require any additional HipLink module (like the CLI or COM object).

The HTTP Interface is a CGI executable, action.exe, that can be invoked using the HipLink url: http://marindrive:8000/cgi-bin/action.exe

Syntax

The following fields are used to send a message to HipLink using the HTTP Interface:

Name	Description
cmd=Cli	HTTP interface command, mandatory value ‘Cli”
modid=cli	Module ID, mandatory value ‘cli’
name=	Recipient name (Receiver name or Group name)
rcv_name=	Receiver name
grp_name=	Receiver Group name
duty_name=	On-Duty Group name
casc_name=	Escalation Group name
rota_name=	Rotate Group name
foll_name=	Follow-me group name
message=	The body of the text message, mandatory

©HipLink Software 2013. All Rights Reserved.

Name	Description
priority=	Message priority, optional from 1 to 5
nametime=	Append name and time to the message, optional (on|off)
carrier_name=	Carrier name, optional, for Quick Send only.
pin=	Receiver PIN, optional, for Quick Send only.
sender=	Custom sender name, optional
jobid=	Custom message job ID, optional
email_subject=	e-mail subject field, optional, for SMTP messages only.

Examples

Send a message to the Receiver 'dtmf-voice'

http://marindrive:8000/cgi-bin/action.exe?cmd=Cli&modid=cli&rcv_name=dtmf-voice& message=test

Send a message to the recipient 'grp-1' (in this case, a Receiver Group)

http://marindrive:8000/cgi-bin/action.exe?cmd=Cli&modid=cli&name=grp-1&message=test

Send a message **using Quick Send**

http://marindrive:8000/cgibin/action.exe?cmd=Cli&modid=cli&carrier_name=smtp&pin=123

Java Interface

Introduction

The HipLink Java API provides HipLink Users with a Java interface for sending wireless messages.

Through this interface, which is similar to the CLI interface and the COM interface, the HipLink product has the ability to receive messages and send them to existing Receivers and/or Groups. The HipLink Java API can be integrated with other customer applications.

The Java API also connects to the HipLink server over HTTP. All parameters (e.g., HipLink IP address or hostname) should be passed as properties of the object. The Java API does not use any config file and does not generate a log file. Error codes and messages are provided by the methods.

©HipLink Software 2013. All Rights Reserved.

Installation and Prerequisites

Installation

The HipLink Java API can run on any machine that can see the HipLink server over IP. It requires some configuration of the HipLink server. See the Prerequisites section.

Download

Download the HipLink COM Interface from the Hiplink site. Please contact Technical Support for assistance/information with the current download links for the HipLink COM Interface.

Prerequisites

HipLink Java API User

For HipLink, the Java API acts like a non-GUI client. For this reason, a Non-GUI User account has to be created for the Java API (i.e., a User of type Non GUI or both that has an IP address - here the IP address of the machine where the HipLink Java API is installed). See the Installation and Administration Guide for details about creating a User.

Note: Only one Non-GUI User can be assigned to an IP address. This User allows running any Non-GUI HipLink interfaces (i.e., CLI, COM Interface, Java API or E-mail Gateway) from a specific server. In the Reports, all messages will appear as sent by the same User (i.e., there is no distinction made for the different Non-GUI interfaces).

HiplinkApi.jar

Fields

public static final int MSG_TYPE_QUICK
public static final int MSG_TYPE_QUERY

Constructor

HiplinkApi

public HiplinkApi(java.lang.String hiplinkURL)
throws java.lang.Exception, java.net.MalformedURLException

Builds a HiplinkApi object and sets the data for future use. Parameters:

hiplinkURL - the URL of the HipLink for Sending Messages. For example:
"http://127.0.0.1:8000".public

Methods

resetParams

public void resetParams()

Clears and resets all the parameters. The URL remains as is, however.

©HipLink Software 2013. All Rights Reserved.

setMsgType

public void setMsgType(int msgType)

throws java.lang.Exception

Sets the type of the message. The various types of messages are provided as public static integers in this class. For example this.MSG_TYPE_QUICK will send a quick message to a Receiver. Depending on which message type is chosen, further parameters are required to send a message.

Throws: java.lang.Exception – if the message type does not exist.

getMsgType

public int getMsgType() Returns the current message type.

setRecipientName

public void setRecipientName(java.lang.String recipientName)

Sets the recipient name (i.e., Receiver or Group).

Note: The recipient must match one of the recipients defined in the HipLink database.

getRecipientName

public java.lang.String getRecipientName() Returns the current recipient name.

setReceiverPin

public void setReceiverPin(java.lang.String receiverPin) Sets the Receiver PIN.

This is only used if the message type is MSG_TYPE_QUICK.

getReceiverPin

public java.lang.String getReceiverPin()

Returns the current Receiver PIN.

setCarrierName

public void setCarrierName(java.lang.String carrierName)

Sets the Carrier name. This is only used if the message type is MSG_TYPE_QUICK. This must be one of the Carriers defined in the HipLink database (created using GUI). Must be followed by the PIN option.

getCarrierName

public java.lang.String getCarrierName() Returns the current Carrier name.

setMessage

public void setMessage(java.lang.String newMessage)

©HipLink Software 2013. All Rights Reserved.

Sets the text of the message. This is mandatory for all message types except for a query.

getMessage

public java.lang.String getMessage() Returns the current text of the message.

getResultText

public java.lang.String getResultText()

In the case of an error, this result text will contain the error description.

getResultCode

public int getResultCode()

200 is success (as with HTTP), everything else is a failure of some sort. See the HipLink Programmer Guide (CLI section) for details.

getJobId

public java.lang.String getJobId()

This will return the current Job Id. After sending a successful message, the Job Id that is generated by the HipLink server will be returned to this API and will be available via this method. The Job Id parameter may be overwritten with setJobId().

setJobId

public void setJobId(java.lang.String jobId)

This sets the job id for use in queries about message status. It must be used after setting the message type to MSG_TYPE_QUERY.

If used while sending, the Job Id overwrites the one generated by the HipLink server and which is guaranteed to be unique. If this feature is used, the User has the responsibility for the Job Id uniqueness.

setUseHighPriority

public void setUseHighPriority(boolean useHighPriority)

Set to true to use high priority, set to false (the default) to use normal priority.

isHighPriority

public boolean isHighPriority()

Returns the current value of this flag.

setUseTimeStamp

public void setUseTimeStamp(boolean useTimeStamp)

©HipLink Software 2013. All Rights Reserved.

Set this to true (the default) to add a time stamp to the message, set to false to not use one.

isUsingTimeStamp

public boolean isUsingTimeStamp()

Returns the current value of this flag.

addRespAction

public void addRespAction(java.lang.String responseText, java.lang.String actionName, boolean isFinal)

This sends adds a response action to a message. Used only for the two-way (SNPP Level 3) messages. Parameters:

responseText - what the recipient(s) will see on the device

actionName - Defines an action to be executed upon receiving a response linked to it. It must match to the predefined action name, defined in the HipLink database (created using GUI).
isFinal - ends further actions for this Two-way message if true.

send

public boolean send()

throws java.lang.Exception

This sends the message to the HipLink server, and processes the response. The returned boolean value reflects whether or not the request was a success (return code 200).

If the send was a success (return value = true) then the results can be obtained the following functions:

- getResultText()
- getJobId()

If the send was a failure (return value = false), then the error code and error text are available using:

- getResultCode()
- getResultText()

Throws: java.lang.Exception - if there is a mistake in the settings (for example if the message text is empty).

Examples for Developers

Example 1: Sending a Message to a Receiver or a Group

HiplinkApi theAPI = new HiplinkApi ("http://192.168.201.46:8000/cgi-bin/ action.exe");
theAPI.setMessage("This is a test message to Dave"); theAPI.setRecipientName("Dave Moss");
if(theAPI.send())

©HipLink Software 2013. All Rights Reserved.

```
{
System.out.println("This job ID is :"+theAPI.getJobId());
}
else
{
System.out.println("err code:"+String.valueOf(theAPI.getResultCode())); System.out.println("err
description:"+theAPI.getResultText());
}
```

Example 2: Sending a Quick Message

```
HiplinkApi theAPI = new HiplinkApi ("http://192.168.201.46:8000/cgi-bin/ action.exe");
theAPI.setMsgType(HiplinkApi.MSG_TYPE_QUICK); theAPI.setMessage("This is a test
message to Lachlan"); theAPI.setCarrierName("smtp");
theAPI.setReceiverPin("lpaterson@hiplink.com");
if(theAPI.send())
{
System.out.println("This job ID is :"+theAPI.getJobId());
}
else
{
System.out.println("err code:"+String.valueOf(theAPI.getResultCode())); System.out.println("err
description:"+theAPI.getResultText());
}
```

Example 3: Sending a Query For Message Status

```
HiplinkApi theAPI = new HiplinkApi ("http://192.168.201.46:8000/cgi-bin/ action.exe");
theAPI.setMsgType(HiplinkApi.MSG_TYPE_QUERY);
theAPI.setJobId("HPG33");
if(theAPI.send())
{
System.out.println("job status is:"+theAPI.getResultText());
}
else
{
System.out.println("err code:"+String.valueOf(theAPI.getResultCode())); System.out.println("err
description:"+theAPI.getResultText())
}
```

©HipLink Software 2013. All Rights Reserved.

SNMP Monitoring

The SNMP Monitoring feature depends on the License Key and provides the HipLink administrators with a tool that monitors the SNMP messages from the local network where the HipLink is installed. The SNMP Monitor is able to generate Alerts based on Events associated with specific Devices connected to the network.

The SNMP Monitor is a HipLink service that can be started and stopped from the Services menu (see the *HipLink Installation and Administration Guide* for details in starting services.)

Note: You need to make sure that there is no other SNMP trap running on the HipLink server.

Devices

To add a new Device:

1. From the Settings menu, click SNMP on the Monitoring section of the left navigation bar.
2. On the “SNMP Device Settings” panel, click the Add Device button to reach the Edit Device Settings page.
3. Enter a unique Device Name for this device (required).
4. Enter a unique Device IP for this device (required).
5. Enter a Description for this device (optional).
6. Click the Save button.

Example: Device Parameters

Name	Value	Comment
Device Name	Ethernet	mandatory
Device IP	123.123.12.3	mandatory
Description	Ethernet Interfaces and Hub	optional

Events

To add an Event to a Device:

1. From the Settings menu, click SNMP on the Monitoring section of the left navigation bar.
2. On the “SNMP Device Settings” panel, find the Device to which you want to add an Event and click the Events icon.
3. On the “SNMP Event Settings” panel, click the Add Event button to reach the Add Event page.
4. Enter a unique Event Alias for this event (required).
5. Enter a unique Event OID for this event (required). When entering Hiplink Event OID all OIDs must begin with . (dot) even if the event OID wasn't sent with a “.” prefix. For

©HipLink Software 2013. All Rights Reserved.

example, 1.3.6.1.4.1.2854 will become .1.3.6.1.4.1.2854; this would have to be entered in the Event OID field under SNMP monitor configuration.

Furthermore, Hiplink SNMP will add .0.x to each Received Trap where x is the specific trap code. (Hiplink uses trap code 1 if no other code is specified.) For example, for an event OID 1.3.6.1.4.1.2854, we would create an event in Hiplink with the following information:

OID

.1.3.6.1.4.1.2854.0.1

Notice the dot at the start. Also, .0.1 has been added to Hiplink Event OID when specifying OID even though they were not in the original event.

Note: Only Enterprise Specific Traps are supported. Cold Start, Warm Start, Link Up, Link Down, Authentication and Neighbor Loss Traps will result in Unknown OID

6. Enter a Description for this event (optional).
7. Enter a Message for this event (required).
8. Click the Save button.

Example: SNMP Event Details

Name	Value	Comment
Event Alias	Authentication	mandatory
Event OID	.1.3.6.1.6.3.1.1.5.5	mandatory
Description	Authentication Failure	optional
Message	Cannot authenticate Ethernet device $H ($I)	mandatory

To import Events to a Device from a MIB file:

1. From the Settings menu, click SNMP on the Monitoring section of the left navigation bar.
2. On the “SNMP Device Settings” panel, find the Device to which you want to add an Event and click the Events icon.
3. On the “SNMP Event Settings” panel, select an MIB file from the dropdown list and click the Import MIB button.
4. The Events defined in the MIB file will be automatically imported.
5. Click the Save button.

SNMP Event Alerts

To set an SNMP Event Alert:

1. From the Settings menu, click SNMP on the Monitoring section of the left navigation bar.

©HipLink Software 2013. All Rights Reserved.

2. On the “SNMP Device Settings” panel, find the Device to which you want to add an Event Alert and click the Events icon.
3. On the “SNMP Event Settings Panel,” find the Event to which you want to add an Alert and click the Alerts icon.
4. On the “SNMP Event Alert” panel select All or a specific Event from the dropdown list of Events.
5. Wait until the Event Parameters are displayed at the top of the page and then you can start to select a list of Recipients.
6. Highlight one or more Receivers and/or Groups and click the Add button to create your Recipient List. You can also add/remove a recipient to/from a list by double clicking on that recipient.

   Note: Acronyms prefix the Recipient names: none, (Num), (2Way), and (Fax) for Receivers, (G) for Receiver Groups, (O) for On-Duty Groups, (E) for Escalation Groups, (R) for Rotate Groups, and (F) for Follow-Me Groups.

7. For managing a large number of Receivers and Groups use either the simple Alphabetical Filtering feature (e.g., by pressing one of the links A, B, C,..., Z, and Others), or the advanced Search feature to display only the information that you need (e.g., enter a keyword followed by a “*” and press the View icon). Note that if you already had defined a list of Personal Favorites, then the content of this list is displayed by default on the left column. Select the link All to display all the available Receivers and Groups, and select the Favorites link to go back and display only those selected in your Favorites List.
8. To select multiple items on a list, click the left mouse button while holding down either the Shift or Ctrl key.
9. Highlight a Group and click the Show Group Members link to see the Members of that Group in a pop-up window. This is useful if you want to see who is on duty right now.
10. Click the Save button to submit your settings and return to the SNMP Event Settings Panel, Reset button to fill in the previous values, or Cancel button to return without saving.

SNMP Monitoring Setup Parameters

Note: There are many traps that are not supported. Currently, Enterprise Level traps are supported with minor settings. Non-Enterprise Level traps require the User to setup the parameters prior to setting any traps and events.

SNMP Traps Supported:

1. Enterprise Level Traps
   a. Any application that may generate a trap.
2. Non-Enterprise Level Traps
   a. DISMAN-EVENT-MIB
      i. authenticationFailure (.1.3.6.1.6.3.1.1.5.5)
      ii. coldStart (.1.3.6.1.6.3.1.1.5.1)
      iii. mteEventSetFailure (.1.3.6.1.2.1.88.2.0.5)

©HipLink Software 2013. All Rights Reserved.

iv. mteTriggerFailure (.1.3.6.1.2.1.88.2.0.4)
v. mteTriggerFalling (.1.3.6.1.2.1.88.2.0.3)
vi. mteTriggerFired (.1.3.6.1.2.1.88.2.0.1)
vii. mteTriggerRising (.1.3.6.1.2.1.88.2.0.2)
viii. warmStart (.1.3.6.1.6.3.1.1.5.2)

b. DISMAN-SCHEDULE-MIB.txt
   i. schedActionFailure (.1.3.6.1.2.1.63.2.0.1)

c. DISMAN-SCRIPT-MIB.txt
   i. smScriptAbort (.1.3.6.1.2.1.64.2.0.1)
   ii. smScriptException (.1.3.6.1.2.1.64.2.0.3)
   iii. smScriptResult (.1.3.6.1.2.1.64.2.0.2)

d. IF-MIB.txt
   i. authenticationFailure (.1.3.6.1.6.3.1.1.5.5)
   ii. coldStart (.1.3.6.1.6.3.1.1.5.1)
   iii. linkDown (.1.3.6.1.6.3.1.1.5.3)
   iv. linkup (.1.3.6.1.6.3.1.1.5.4)
   v. warmStart (.1.3.6.1.6.3.1.1.5.2)

e. IPV6-MIB.txt
   i. ipv6IfStateChange (.1.3.6.1.2.1.55.2.0.1)

f. NOTIFICATION-LOG-MIB.txt
   i. None

g. RFC-1215.txt
   i. None

h. RMON-MIB.txt
   i. fallingAlarm (.1.3.6.1.2.1.16.0.2)
   ii. risingAlarm (.1.3.6.1.2.1.16.0.1)

i. SNMPv2-MIB.txt
   i. authenticationFailure (.1.3.6.1.6.3.1.1.5.5)
   ii. coldStart (.1.3.6.1.6.3.1.1.5.1)
   iii. warmStart (.1.3.6.1.6.3.1.1.5.2)

j. SNMPv2-SMI.txt
   i. None

k. UCD-SNMP-MIB.txt
   i. StartucdShutdown (.1.3.6.1.4.1.2021.251.2)
   ii. ucd (.1.3.6.1.4.1.2021.251.1)

HipLink Setup for Supported Traps:

1. Enterprise Level Traps (Manual traps)
   a. Creating a new SNMP device, enter the IP of machine(s) from which events would be generated. (Server IP where application is installed, 192.168.4.81).
   b. Create Event against the new SNMP Device.
   c. When entering HipLink Event OID, all OID's must begin with. (dot) Event if the event OID wasn't sent with a "." prefix. For example, 1.3.6.1.4.1.2854 will become

©HipLink Software 2013. All Rights Reserved.

.1.3.6.1.4.1.2854 but you would have to Enter the "." yourself in the Event OID field under SNMP monitor configuration. Otherwise it will not work.

d. Append *.0.<specific trap type code>* in the OID to allow matching. e. A device has unique Event names.
e. Recipients are associated with events. The event message is delivered to the Recipient
f. List on Event fire.

2. Non-Enterprise Level Traps
    a. Creating a new SNMP device, enter the IP of machine(s) from which events would be generated. (Server IP where application is installed, 192.168.4.81).
    b. Select the MIB (Management Information Base) file having desired events and press
    c. "Import MIB" button.
    d. Populate the recipient list by setting up alerts.

Why some OID's are different than expected:

SNMP Traps have generic trap types and specific trap types. Standard generic traps are: coldStart, warmStart, linkDown, linkUp, authenticationFailure, egpNeighborLoss. For vendor specific traps, Generic trap type field is set to enterpriseSpecific(6). Specific trap indicates one of a number of specific trap codes, used when generic trap type is set to enterprise. The specific trap type further classifies the trap if in case the trap itself is not standard.

Enterprise traps are differently processed due to a reason that their OIDs represent vendor specific system and appending a .0.<specific_type> will transform the incomplete OID into that vendor specific system's object OID. Thus, if an enterprise trap is caught, then it is the agent's responsibility to generate the correct OID by appending the .0 if not in the end of the OID and then the specific type code. The resultant OID becomes the complete OID that will represent the object in the vender specific system. However for other traps (when generic trap is other than enterprise) the auto appending of .0.<specific_type> is not done and neither required as the OID in the trap completely identifies the event/object in the target.

Troubleshooting Steps:

1. Restart SNMP services from HipLink services panel.
2. Set up network connection properly
3. Restart Systems SNMP services.
4. System reboot

Additional Information:

To set up System's SNMP service, follow the steps below.

- Go to control panel
- Select Add/Remove Programs
- Select Add/Remove Windows component
- Select Management and Monitoring Tools and press Details buttons
- Select Simple Network Management Protocol
- Press "OK" button

©HipLink Software 2013. All Rights Reserved.

- Press “Next”

- Now, start SNMP service on all the machines to be used
- Go to the Control Panel >Administrator Tools >Services and edit the properties of SNMP service on each machine
- Go to the TRAPS tab and enter the following details
    - Community Name : public

TRAP Destinations: Enter all the IP’s being used for SNMP on each machine being used

FSI Interface

Introduction

The HipLink FSI (File System Interface) provides a powerful and easy way to send messages using HipLink.

As an alternative to using the GUI, CLI, COM, or Java API interfaces, a message can be created as a text file and copied directly into a HipLink predefined spool directory to be processed. This interface gives more flexibility to integrate the HipLink messaging system into various processes and applications.

The FSI may be run on the same machine as the HipLink Server, or it may be run on a different machine. It communicates with the HipLink server using HTTP, so it may also be used on a different platform. Multiple instances of the FSI may be run on multiple machines.

The message file structure is described in the ‘Message File Format’ section below.

Prerequisites

The availability of the FSI interface is based on the License Key. No installation is needed. The FSI is configured using the GUI.

The FSI executable is called hipfi.exe and it is located in the bin directory on all the platforms. Before starting the File System Interface service you should create a User of type Non GUI or both that has the IP address of the HipLink Server.

Starting and Stopping the FSI

If the FSI is being run on the same machine as the HipLink server, then you may use the GUI to start and stop the FSI service. See the User Guide for details concerning starting and stopping services with the GUI.

If the FSI is running on a remote machine then the GUI is not available, but since this is a simple executable, it may be started and stopped in a variety of ways. The FSI requires the use of a config file, which is specified as a command line parameter -config_dir.

- Executed on the Command Line. For example:

©HipLink Software 2013. All Rights Reserved.

hipfi.exe -config_dir /usr/local/hiplink/config

- Executed from any program that has access to the Command Line.
- Startup Script
- Windows Services (control panel->services, then start/stop the HPD-HIPFI service)

Note: on UNIX it will run as a daemon and return the prompt. You can end the FSI process by simply killing it with the standard UNIX “kill” command.

Setting FSI Parameters

Using the GUI

The procedure of setting the FSI parameters using the GUI is briefly explained below. For more details refer to the HipLink User Guide.

If the FSI is enabled, the File Interface entry will be displayed in the System category, in the left navigation bar of the Settings Menu. You can click on the link and go to the File System Interface Panel to view the default settings that you can modify by clicking the Edit button.

The URL used by the FSI Interface is that of the HipLink server (i.e., by default http://127.0.0.1:8000).

On Windows platforms, the default spool directory is:

C:\Program Files\HipLink Software\HipLink\hipfi_spool

On Unix platforms, the default spool directory is:

/usr/local/hiplink/hipfi_spool

Once that is started, the FSI service checks periodically the spool directory to see if there are new files that need to be processed. If there are, it will move them in the HipLink main queue to be processed.

The spool directory checking period is a parameter that can be set by the User in the range 30 to 60 seconds. The default value is 30 seconds.

Like any other HipLink service, the FSI has its own log file. Debugging details in the log file can be toggled on and off from a checkbox. This feature is disabled by default.

Editing the Config File

If the FSI is running on a remote machine, there are two options for setting the parameters in the config file.

1. You can set up the parameters using the GUI on the HipLink Server and then copy this file to the remote machine: /hiplink/config/hipficonfig

or

2. You can also edit the hipficonfig file directly with any text editor. The format of the file is very simple. A Windows example is below. The spaces between the variable names and the values should be TABs and not spaces.

©HipLink Software 2013. All Rights Reserved.

HipLink_url: http://192.168.201.9:8000
spool_dir: C:\Program Files\HipLink Software\HipLink\hipfi_spool
log_dir: C:\Program Files\HIpLInk Software\HipLink\logs logfile_size:
256000
logfile_history: 15
rest: 30
debug: 1

Where:

- HipLink_url - the url of the HipLink Server
- spool_dir - the directory where the FSI will look for new files to process
- log_dir - the directory where the FSI will write logs
- logfile_size - this is the maximum size of the log file in Bytes
- logfile_history - the number of days to keep the log file
- rest - the time in seconds that the process will sleep between subsequent checks of the spool directory
- debug - 1 means true, 0 false - this determines if debug info is written to the log file or not.

Message File Format

The message file format is also very simple:
RECEIVER: receiver_name
GROUP: receiver_group_name
NAME: receiver_name or group_name
PIN: pin
CARRIER: carrier
JOBID: job_id <optional>
MESSAGE: body of the message <required, can be more than 1 line>

Either the RECEIVER, GROUP,NAME, or PIN and CARRIER are required.

Note: Use RECEIVER to send to HipLink Receivers, GROUP to send to Receiver Groups, or NAME to send to Receivers an all types of Groups (Receiver Groups, On-Duty, Escalation, Rotate, or Follow-Me Groups).

The parameters that are not used have to be skipped.

In order to be processed by the FSI service, the message file name should start with “hipfi”. The file extension is optional. After they are processed, the files are automatically deleted from the spool directory.

Example 1

The content of the file hipfi_FSI_1002 is:

RECEIVER: John Doe
JOBID: FSI_1002
MESSAGE: Hello John, the weekly meeting was rescheduled for Thursday at 2:00pm.

Example 2

©HipLink Software 2013. All Rights Reserved.

The content of the file hipfi1016.txt is:

PIN: 1839
CARRIER: SkyTel
JOBID: FSI_1016
MESSAGE: Don't forget to change the backup tape before you leave.

Custom Response Actions

Response Actions allow a User or a receiver to execute system commands remotely. Users can execute response actions in two ways:

- Through the HipLink Mobile Client (ActionGUI)
- By associating a response action to an event on Feedback Action Panel Receivers can execute response actions from a two-way wireless device. There are two ways to do this:
  - when sending a two-way message via Two-Way Send panel, Users can add response actions to the messages.
  - while creating a two-way carrier (for OAI, VoIP protocols), Users can associate response actions to the *Response Text*, the answer options created for a two-way message.

When a two-way receiver responds to the message by selecting a response text, the command is executed.

Response Actions panel is available to all sysAdmin Users. For non-sysAdmin Users, it is available only if add, modify, or delete Response Actions permission is given in User Group's settings. The panel can be accessed from Settings tab's left navigation bar in General menu, as well as from the main Settings panel.

System Response Actions

There are six pre-defined Response Actions in HipLink, which appear on Response Actions panel with a View icon. These response actions can neither be modified nor deleted:

Conf: to confirm a message

GetUserData: to get the User information

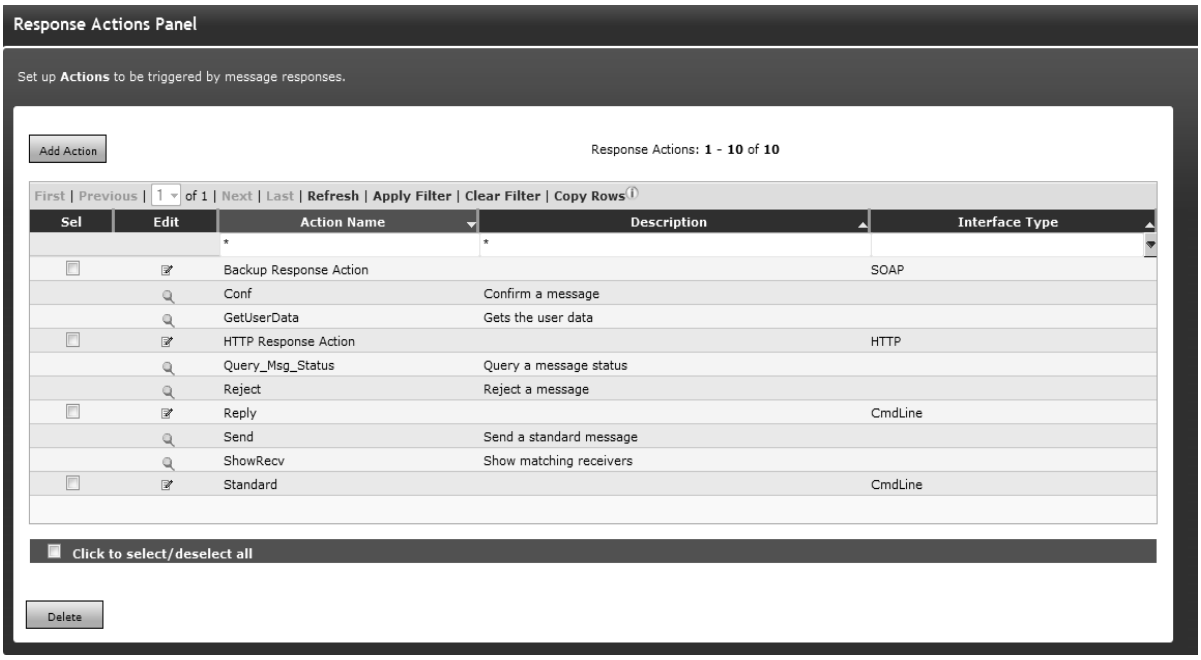
Query_Msg_Status: to get the status of a message

Reject: to reject the message

Send: to send a standard message

ShowRecv: to show receivers matching the given receiver pattern

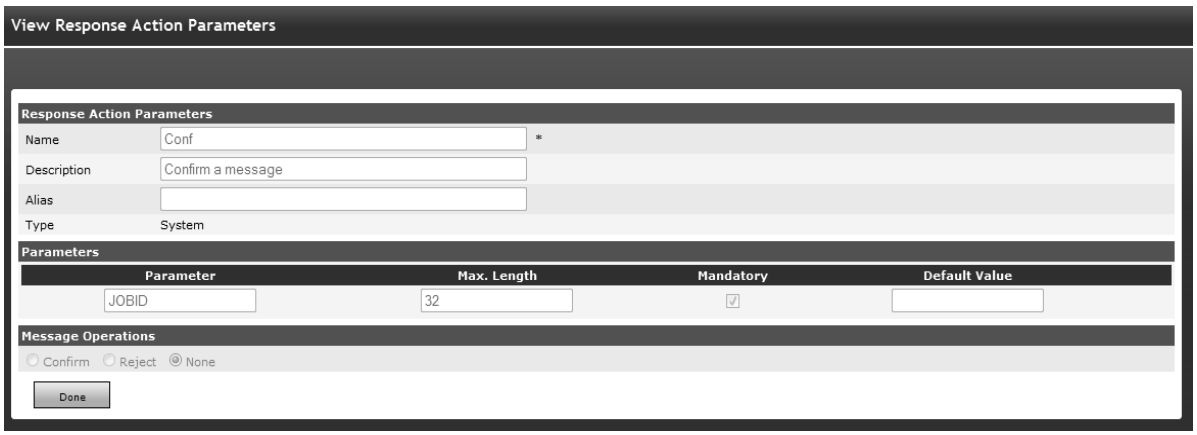
©HipLink Software 2013. All Rights Reserved.



Response Actions Panel

The parameters required by these response actions are explained below:

1. Conf:



View Response Action Parameters Conf

JOBID: ID of the HipLink job to be confirmed. This is a string with a maximum length of 32 characters (mandatory).

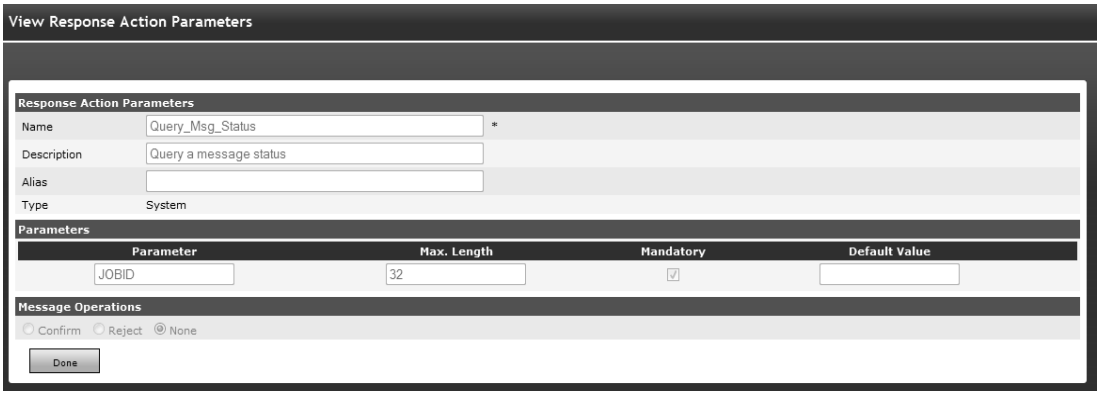
2. GetUserData:

©HipLink Software 2013. All Rights Reserved.

View Response Action Parameters GetUserData

NAME: Name of the HipLink User whose data is to be fetched. This is a string with a maximum length of 50 characters (mandatory).

3. Query_Msg_Status:

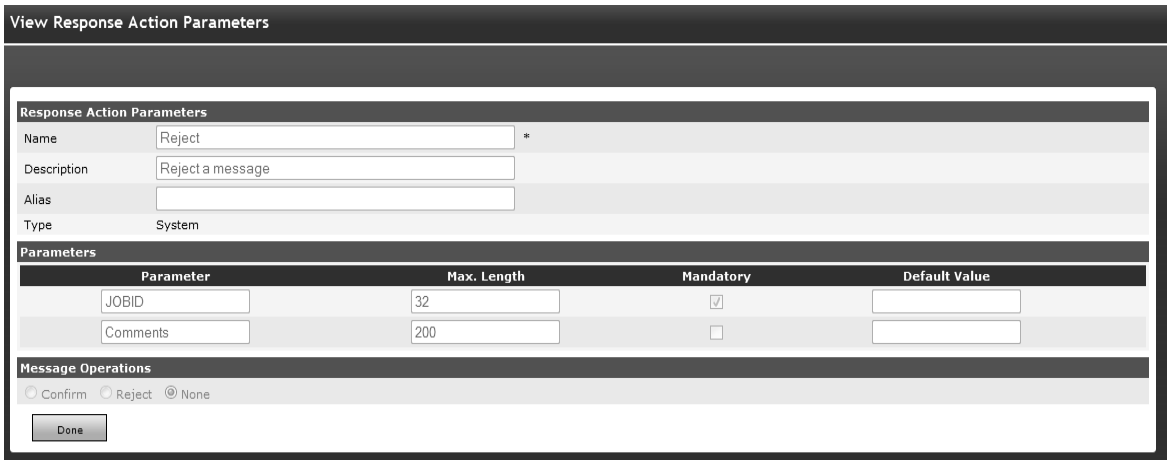


View Response Actions Parameters Query_Msg_Status

JOBID: ID of the HipLink job whose status is queried. This is a string with a maximum length of 32 characters (mandatory).

4. Reject:

©HipLink Software 2013. All Rights Reserved.

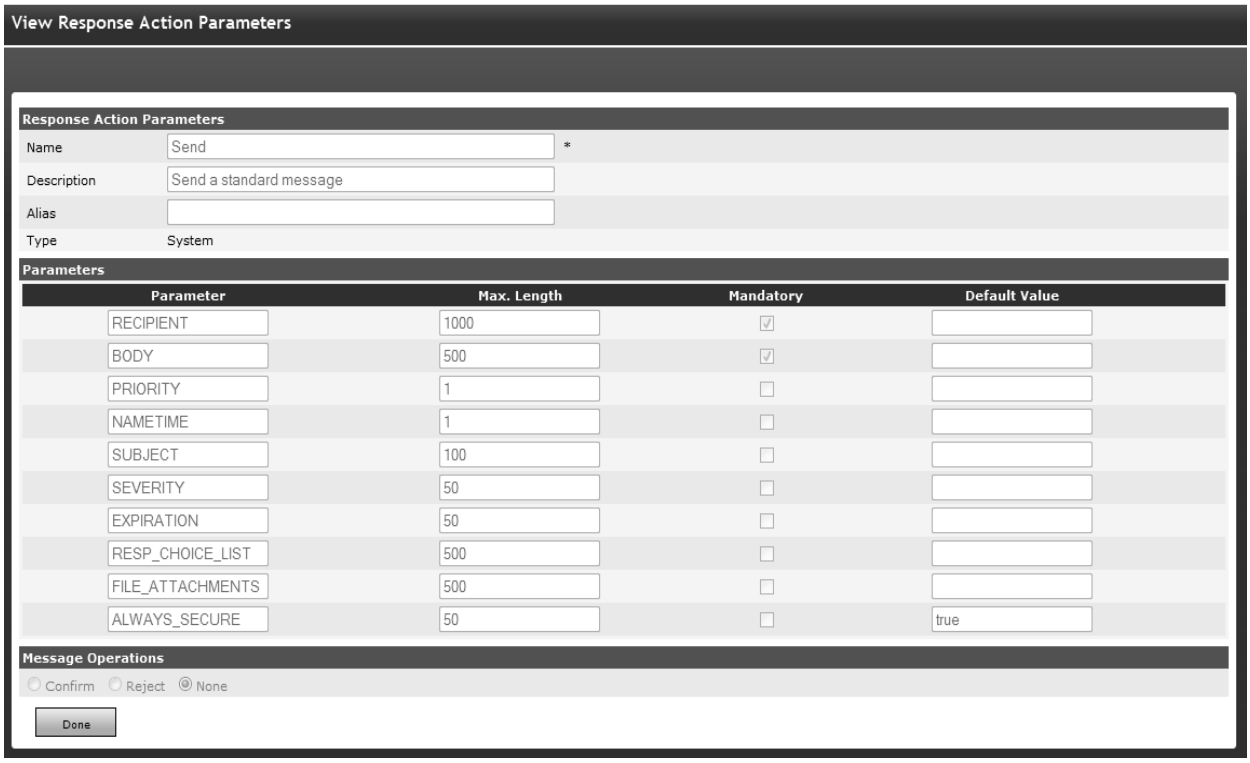


View Response Actions Parameters Reject

JOBID: ID of the HipLink job that you want to reject. This is a string with a maximum length of 32 characters (mandatory).

Comments: Any comments to be given when rejecting the job. This is a string with a maximum length of 200 characters (optional).

5. Send:



View Response Actions Parameters Send

©HipLink Software 2013. All Rights Reserved.

RECIPIENT: The message recipient. This is a string of maximum 50 characters (mandatory).
BODY: The message body. This is a string of maximum 500 characters (mandatory).
PRIORITY: The priority of the message. This is a string of 1 character (optional).

NAMETIME: The name of the sender and the time stamp to be included in the message body. This is a string of 1 character (optional).

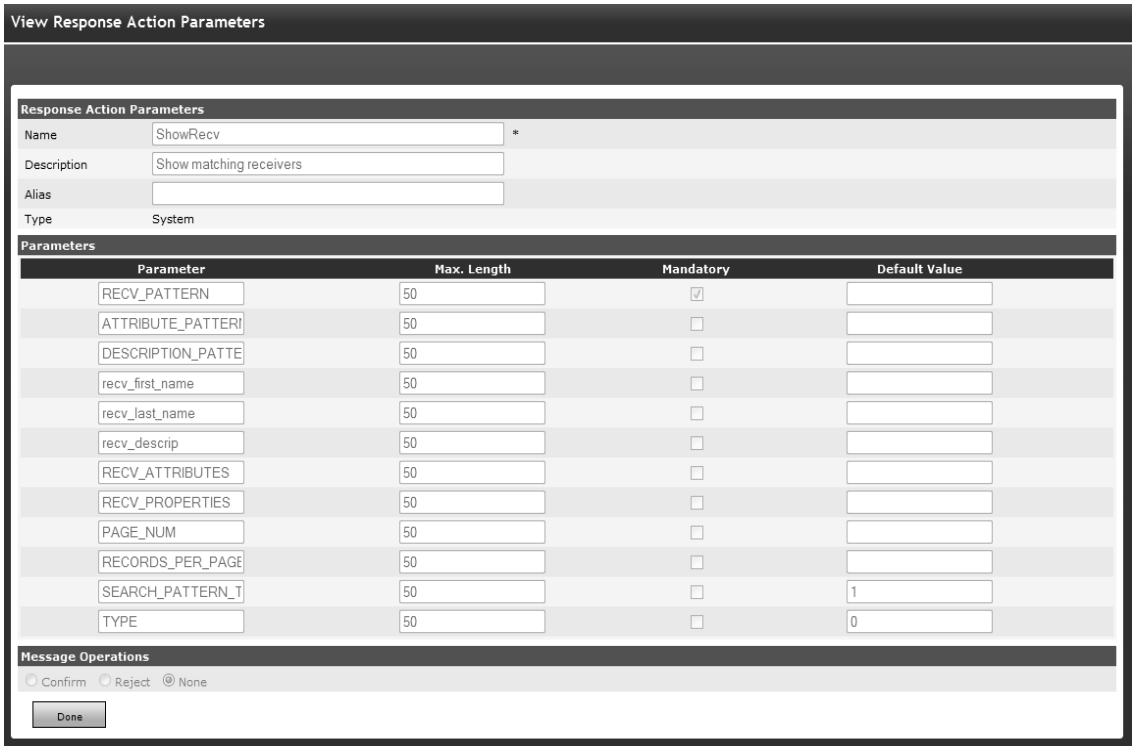
SUBJECT: The subject of the message. This is a string of maximum 100 characters (optional).
SEVERITY: The severity of the message. This is a string of maximum 50 characters (optional).
EXPIRATION: The expiration time of the message, i.e. the time after which message will not be readable. This is a string of maximum 50 characters (optional).

RESP_CHOICE_LIST: The list of response actions to be executed with two way messages. This is a string of maximum 500 characters (optional).

FILE_ATTACHMENTS: The file attachment option. This is a string of maximum 500 characters (optional).

ALWAYS_SECURE: This is a string of maximum 50 characters (optional).

6. ShowRecv:



View Response Actions Parameters ShowRecv

RECV_PATTERN: The pattern of receiver name on which search would be made. This is a string of maximum 50 characters (mandatory).

ATTRIBUTE_PATTERN: The pattern of receiver attribute on which search would be made. This is a string of maximum 50 characters (optional).

©HipLink Software 2013. All Rights Reserved.

DESCRIPTION_PATTERN: The pattern of receiver attribute on which search would be made. This is a string of maximum 50 characters (optional).

recv_first_name: To display receiver's first name on the output page. This is a string of maximum 50 characters (optional).

recv_last_name: To display receiver's last name on the output page. This is a string of maximum 50 characters (optional).

recv_descrip: To display receiver description on the output page. This is a string of maximum 50 characters (optional).

RECV_ATTRIBUTES: To display receiver attributes on the output page. This is a string of maximum 50 characters (optional).

RECV_PROPERTIES: To display receiver properties on the output page. This is a string of maximum 50 characters (optional).

PAGE_NUM: To display records from the given page number (i.e., Records on the specific page will be displayed. This is a string of maximum 50 characters (optional).

RECORDS_PER_PAGE: Number of records to be displayed per page. This is a string of maximum 50 characters (optional).

SEARCH_PATTERN_TYPE: The type of search pattern to be used. This is a string of maximum 50 characters (optional). By default, this parameter is on (value=1).

TYPE: To display receiver's type on the output page. This is a string of maximum 50 characters (optional). By default, this parameter is off (value=0).

Add Response Actions:

Along with the six pre-defined system response actions, Users can define their own response actions and add them to two-way messages or feedback actions. These custom response actions appear on Response Actions panel with Edit icon.

To add a response action, click on Add Action button on the panel. This will direct User to Add Response Action Parameters page.

There are four sections on Add Response Action Parameters page:

- **Response Action Parameters**: Common to all response actions.
- **Parameters:** Depend upon the Type of Response action.
- **Action:** Common to all response actions.
- **Message Operations:** Available only when the Type of Response action is Reply.

Response Action Parameters:

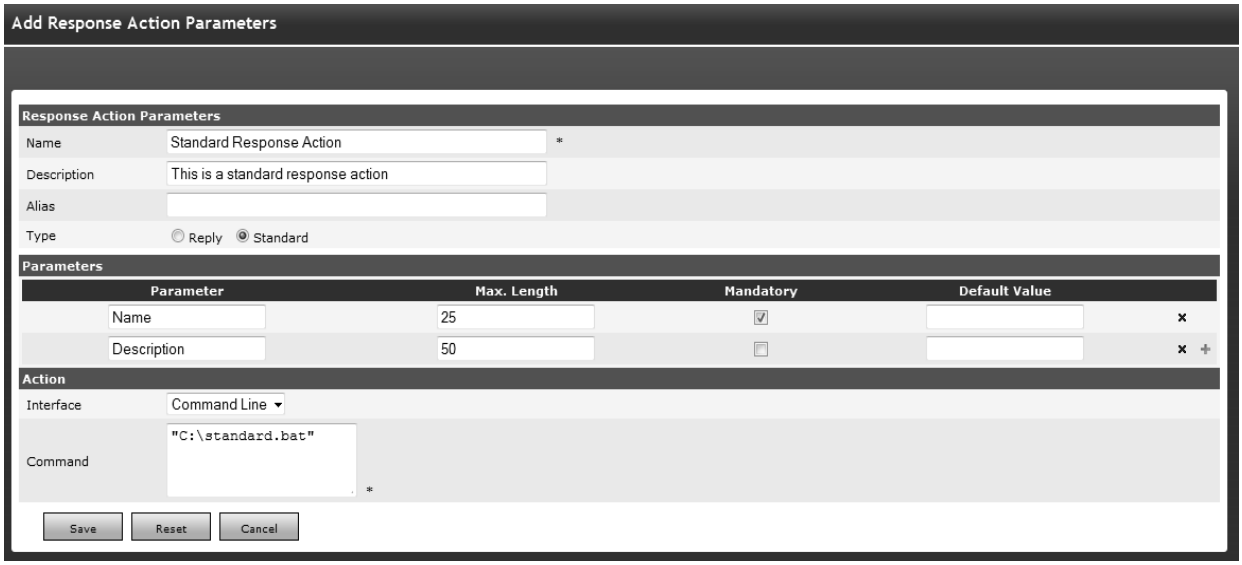
1. Enter a unique **Name** for the response action (mandatory).
2. Enter a **Description** for this response action (optional).
3. Enter an **Alias** for the response action (optional).
4. Select the **Type** of response action (mandatory). By default, the type is selected as Standard.

©HipLink Software 2013. All Rights Reserved.

Parameters:

The fields in this section depend upon the Type of response action selected above.

1. Type: Standard



Add Response Actions Parameters Type: Standard

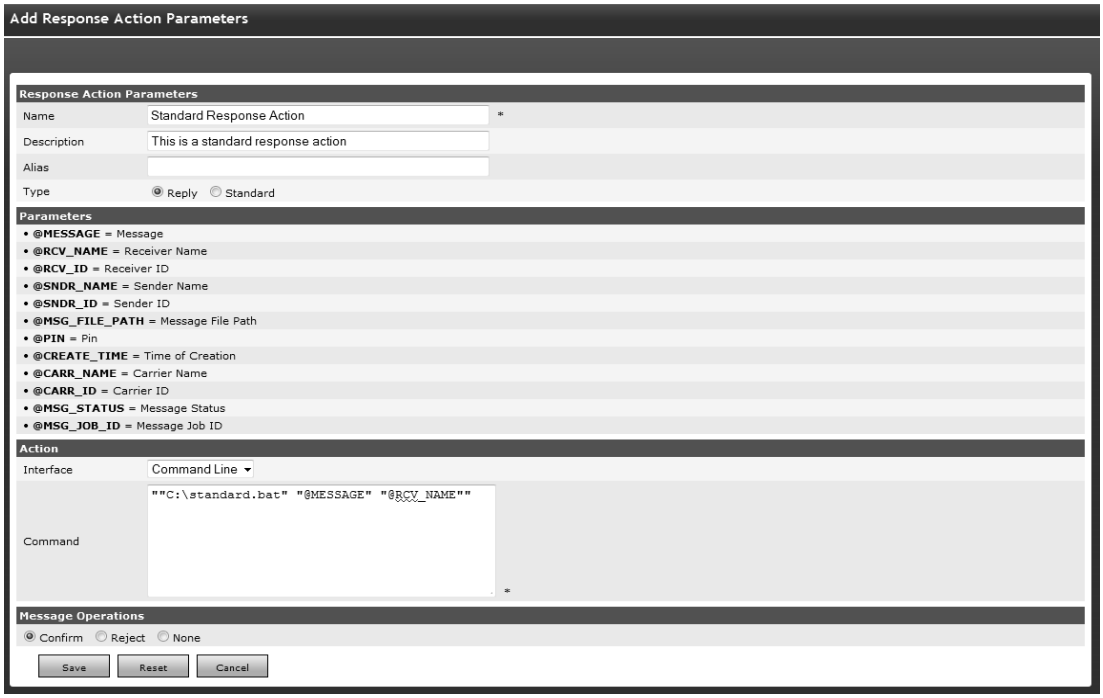
For Standard type response action, following fields are available:

1. **Parameter:** Give the name of the parameter you want to add to this response action.
2. **Max. Length:** Define the maximum length for this parameter.
3. **Mandatory:** Select this checkbox if you want to make this a mandatory parameter. If this checkbox is not selected, the parameter will be optional.
4. **Default Value:** Assign some default value to this parameter. If left blank, the parameter's default value will be set to null.

There is a + icon with the last row added in this section to add a new row of parameters, and an x icon with each row to delete it.

©HipLink Software 2013. All Rights Reserved.

5. Type: Reply

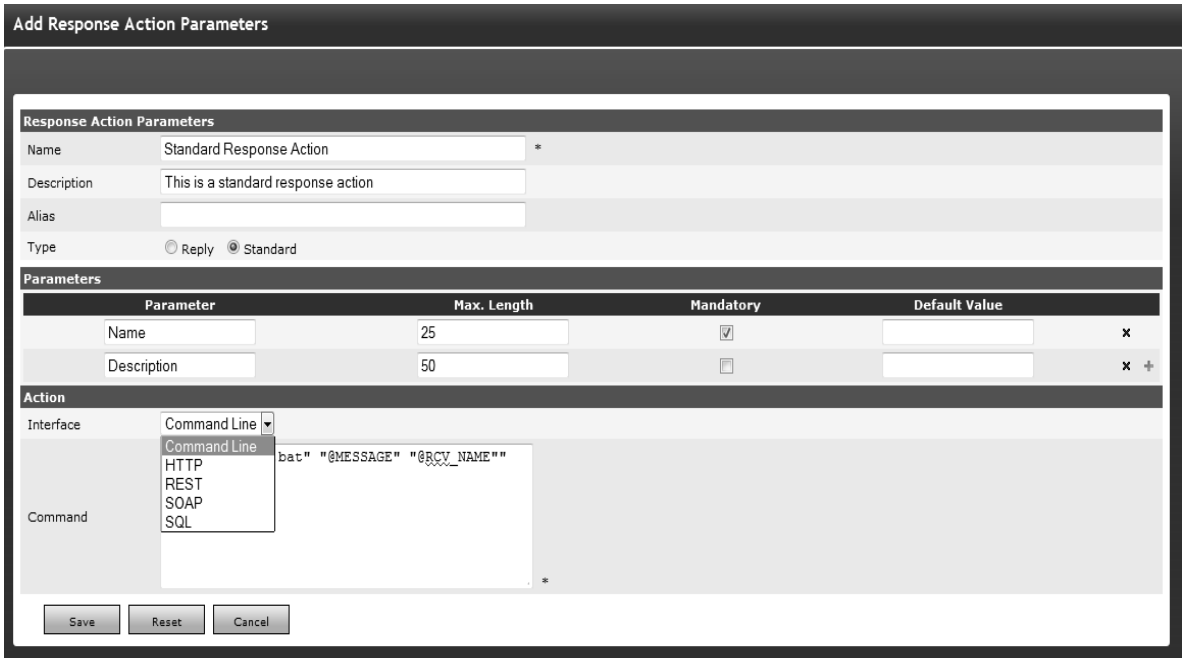


For Reply type response action, there are no fields in Parameters section; only the list of parameters and their descriptions that can be used in the response action.

Action:

1. Select an Interface from the dropdown list (mandatory). There are four interfaces available in this dropdown:
    - Command Line
    - HTTP
    - REST
    - SOAP
    - SQL

©HipLink Software 2013. All Rights Reserved.



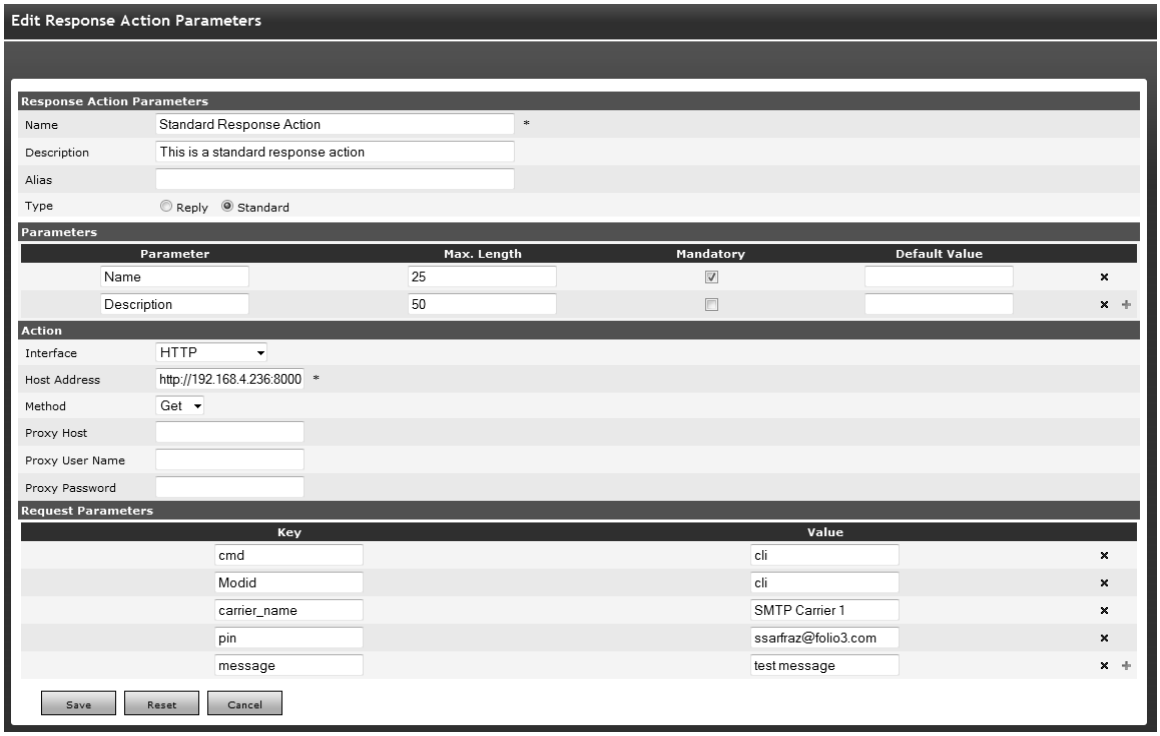
The fields in this section depend upon the interface selected. We will explain each of them in detail here:

1. Command Line:

Command: Enter an executable command accessible from the server where HipLink is installed (mandatory).

2. HTTP:

©HipLink Software 2013. All Rights Reserved.



Host Address: Enter the address of the host http server (mandatory).

Method: Select e method from dropdown list. (Get / Post)

Proxy Host: If any proxy host exists, then enter Proxy address (optional).

Proxy User Name: Enter User name (optional).

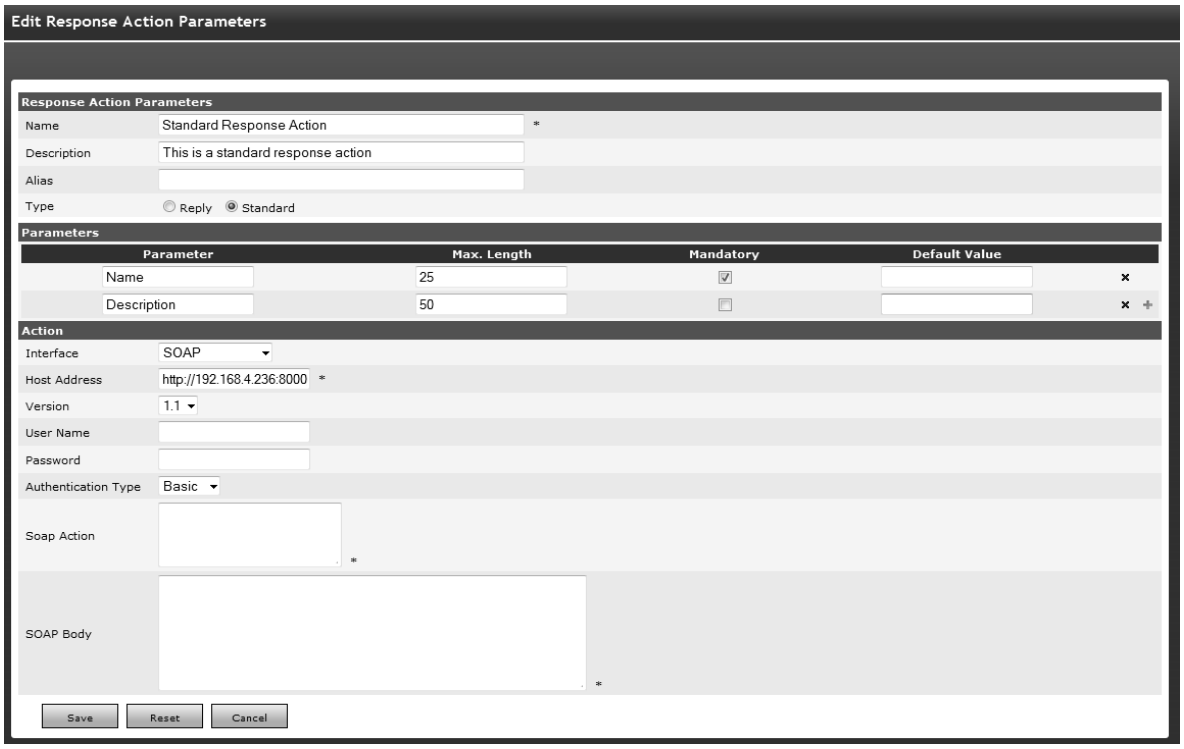
Proxy Password: Enter password (optional).When HTTP is selected as an Interface, a new section Request Parameters is added under this section. Define the parameters to be sent with the http request.

3. REST:

The fields for REST interface are same as those of HTTP.

4. SOAP:

©HipLink Software 2013. All Rights Reserved.



Host Address: Enter the address of the host http server (mandatory).

Version: Select soap version number from the dropdown list. (1.1 /1.2)

User Name: Enter User name (optional).

Password: Enter password (optional).

Authentication Type: Select authentication type from the dropdown list. (Basic / Digest)

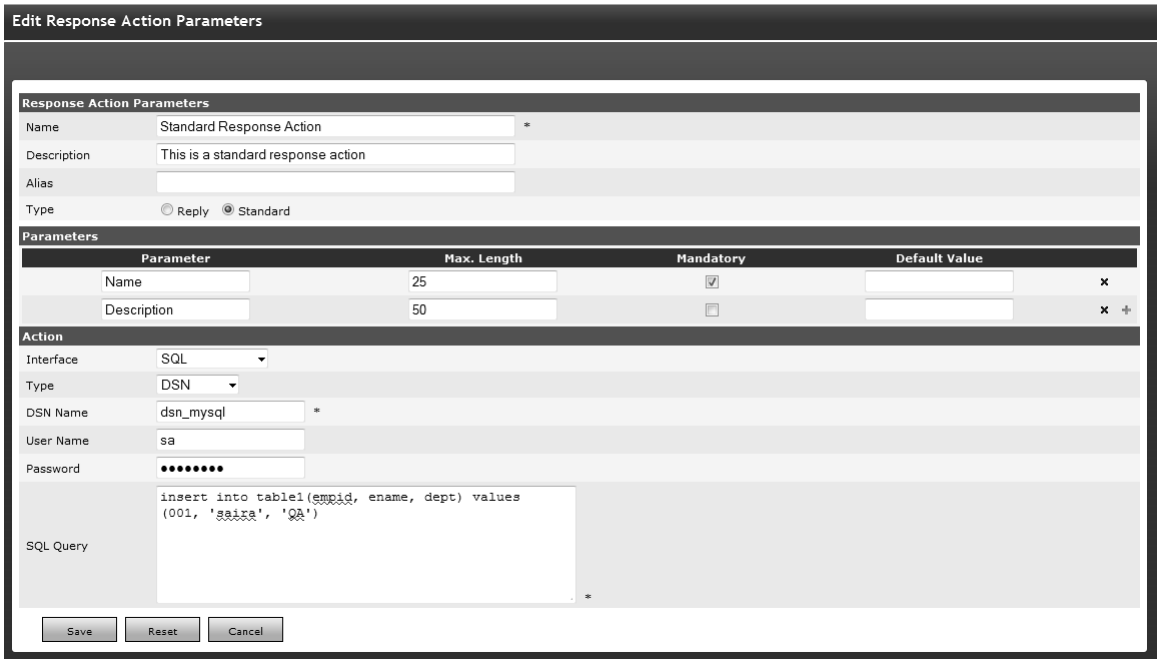
Soap Action: Enter soap action for target method.

Soap Body: Enter soap command which you want to execute

(mandatory).

5. SQL:

©HipLink Software 2013. All Rights Reserved.



DSN Name: Enter a valid DSN name (mandatory).

User Name: Enter a User name for accessing database (optional)

Password: Enter password (optional).

SQL Query: Enter any query which you want to execute (Mandatory*).*

Message Operations:

There are three message operations available in this section:

- **Confirm:** to confirm a message
- **Reject:** to reject a message.
- **None:** to do nothing (No Operation).

Note: When any of the Confirm or Reject options is selected, the response action is not available on Feedback Action panel. When None is selected, the response action is not available on two-way carriers' panels.

Executing Actions Using the ActionGUI

HipLink Generic client – called ActionGUI – allows Users to execute actions using a simple HTML interface by accessing the 'actiongui.exe' application. The URL for this interface is in the following format:

http://<machinename:portnumber>/cgi-bin/actiongui.exe

1. Log in using the User Access Code.

©HipLink Software 2013. All Rights Reserved.

The ActionGUI Login Screen.

2. Upon successful login, the list of available Response Actions will be displayed.

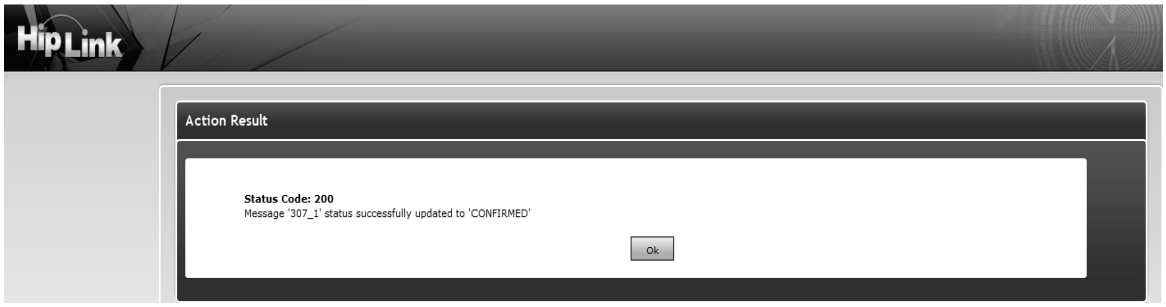
The list of available Response Actions.

3. Select a Response Action by clicking on the Run icon

4. Fill the form with the action parameters and click the Submit button.

5. Get the result.

©HipLink Software 2013. All Rights Reserved.



The Action Result Screen.

6. Click the Ok button.

7. Log off.

Executing Remote Actions Remotely

HipLink allows permissioned Users to execute actions from a remote location in three different ways:

- by using the HipLink Voice Module,
- by using the ActionGUI interface, and
- by sending an email to the Email Gateway.

Permissioned Users

A permissioned User, is a User that beside the password required to log in the HipLink GUI has defined a numeric Access Code. This Access Code is used to authenticate the User while initiating an IVR session from a phone, and it is also entered in the Subject field of an email sent to the Email Gateway.

See the Settings Menu chapter for details about creating a User and setting the Access Code.

Predefined Actions

In the case of an IVR session, the User will be asked to select the action from an option list. For details about installing and using the HipLink Voice Module refer to the HipLink Installation and Administration Guide.

In the case of sending an email to the Email Gateway, the User will have to send the action and its parameters in the body of the email message.

The actions that can be executed are either the predefined action or any custom response actions defined by the HipLink administrator.

©HipLink Software 2013. All Rights Reserved.

A sysAdmin User can execute all the actions defined on Response Actions panel of HipLink. For a non-sysAdmin User, only those actions are available for which the he/she has been given permissions in User Group settings. (See Section 4.1.2: User Groups for further details).

There are six predefined actions in HipLink:

- **Conf:** to confirm a message
- **GetUserData:** to get the User information
- **Query_Msg_Status:** to get the status of a message
- **Reject:** to reject the message
- **Send:** to send a standard message
- **ShowRecv:** to show receivers matching the given receiver pattern

Executing Actions Using the Email Gateway

The Email Gateway can be configured with two email addresses:

- a one-way email address required for sending messages to HipLink Receivers and Groups,
- a two-way email address required to execute HipLink actions.

An Action may or may not return a response to the sender. If the sender is a permissioned User and the action has a response, then the User will receive it in an email sent to his/her email address.

See the Two Way Email Address Configuration settings on the Email Gateway Panel from the Settings Menu chapter for details.

The execution of actions can be controlled at the Email Gateway level by defining two files called allowed and denied containing the access list of allowed and denied email addresses or domains. These files are located in the config directory, they are both in text format, and contain one address per row.

Four situations are possible:

- If both allowed and denied lists are empty, all senders are allowed to execute actions.
- If the allowed list is not empty and denied is empty, then all senders that are on the allowed list will be allowed to execute actions.
- If the allowed list is empty and denied is not, then all senders that are not on the denied list will be allowed to execute actions.
- If there are entries on both the allowed and denied lists, only these email address or domains that are on the allowed list (and that are not on denied list, in this particular order) will be allowed to execute actions.

©HipLink Software 2013. All Rights Reserved.

Feedback Actions

HipLink allows Users to add response actions with Feedback events that will be executed upon the status of a message. The following are the types of status that a message can have which allow feedback events to be executed:

- Message backed-up
- Message completed
- Message confirmed
- Message delivered
- Message filtered
- Message finalized
- Message rejected

Note: Only the admin User Group has permission to set up Feedback Actions, and it is the responsibility of the HipLink administrators to write and maintain the scripts, batch files, or applications invoked as Feedback Actions.

Example: To add a feedback event Final event select this event and press Add. This event will be added as a Feedback Action, now specifies the Response Action(s) to be executed when a message state matches the state of Event.

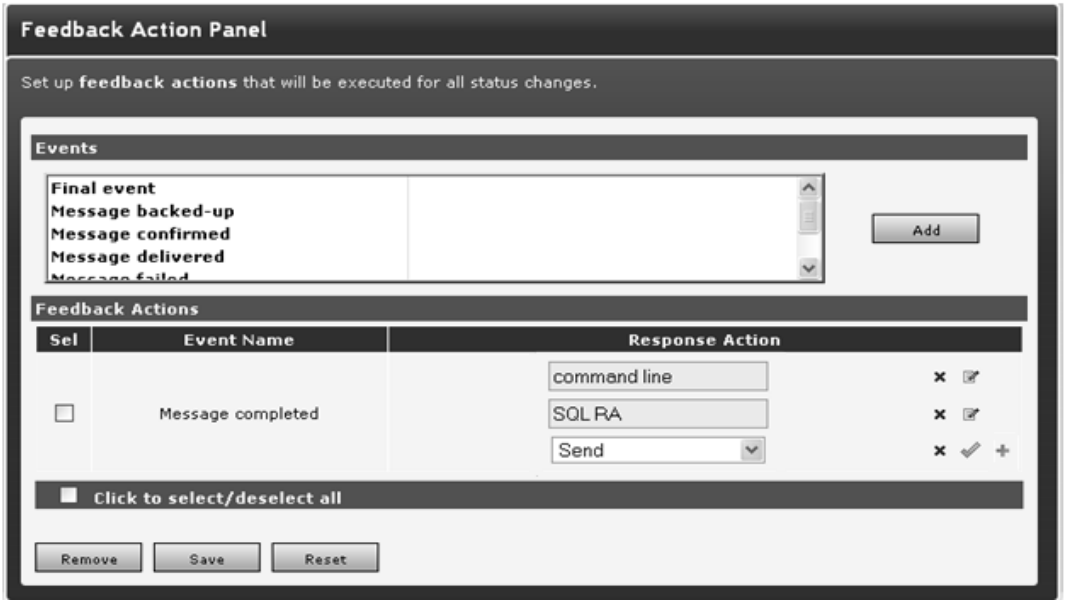
See the Integration Guide for full instructions to define Feedback Events.

Feedback Action Panel

©HipLink Software 2013. All Rights Reserved.

Setting up Feedback Actions:

- From the Settings menu, click Feedback from the System section on the left navigation bar.
- On the Feedback Actions Panel, Select any event and click Add.
- Under Response Action section, select any response action from drop down.
- User can add more than one Response Actions by clicking + sign again each event.
- Click the Save button.



Adding response action within the feedback event.

Note: For more details on Response Action see section 4.5.2

©HipLink Software 2013. All Rights Reserved.

Voice XML

Introduction

The HipLink Voice Module allows a HipLink administrator to receive incoming calls and to send outgoing voice calls. In either case the Voice Module creates the interface to allow Users to control telephony IVR (Interactive Voice Response) session flow, and to perform HipLink functions over the phone. The way to program the IVR interface is by writing an XML document: SVXML (HipLink Voice XML).

SVXML is XML based language designed to control telephony IVR session flow.

SVXML script is set of states, actions, transitions which defines the given IVR session flow (state machine).

SVXML scripts will be interpreted and executed by HipLink IVR server.

SVXML is an XML document that allows a programmer to specify an entire IVR session. IVR describes a computer's ability to interpret Touch Tone or speech input and responds with prerecorded voice over a standard telephone line.

An SVXML document may be created using the text editor of your choice. SVXML scripts will be interpreted and executed by the HipLink Voice Module. The XML files must be stored in the directory specified in the Voice Module GUI.

SVXML describes how to:

- control the dialog flow
- play sound files, dates, numbers, money...
- speak text files, text strings (TTS)
- record sound files
- recognize touch tone key presses (DTMF)
- send a HTTP request to the HipLink server
- control the telephony (make outbound call, hang-up).

An SVXML script can have children SVXML scripts. In this case it is called parent script.

How It Works

One IVR call is represented by one SVXML document. A call contains a set of state elements. Each state element contains one or more action elements and one transitions element. A transitions element is essentially a switch statement that contains cases that capture action exit events and then link to other states. It is a state machine.

The HipLink IVR Module will process the XML document, one state at a time. The first state is identified by its special unique id: STAR" (see the <state> element below for details). The processor will perform the actions of that state one by one until they are done. Each action will end with an event. After all the actions have been performed, the transitions element will be processed. The transitions element is simply a switch statement that contains cases which link the exit events of the actions to other states. The User can hop between states indefinitely. The special Stop action will skip any further transitions and end processing right away.

©HipLink Software 2013. All Rights Reserved.

Note: Both inbound and outbound calls are supported by the same SVXML document specification.

This is the hierarchy of the main elements:

- <?xml version="1.0" standalone='yes'?>
- <call> (1)
- <state> (1..N)
- <action></action> (1..N)
- <transitions></transitions> (1)
- </state> <call>
- <call>

SVXML Elements

<call> Element

A call is the root element for the entire IVR session. It does not have any attributes. Each call should have

element: state [1..N] as described below.

Example

- <call>
- <state> …</state>
- <state>…</state>
- …
- </call>

<state> Element

A state contains two types of elements: action and transitions. It has two attributes ID and name.

Each state should have

- attribute: ID [mandatory]. ID must be unique within the <call> element. The first state ID must be START.
- attribute: name [optional].
- element: action [1..N]. Described below.
- element: transitions [1]. Described below.

An event is the result of action(s) executed at particular state. Events may be triggered by DTMF input or simply a function result. The most recently executed action always overwrites the current event value. For example, if the silence timer runs while a message is being recorded in the action RecordFile, then the event value will be MaxSilence.

Example

- <state id="state3" name="Getting a Password">
- <action>...</action>
- <action>...</action>

©HipLink Software 2013. All Rights Reserved.

- <transitions>...</transitions>
- </state>

<action> Element

Action is a function supported by HipLink Voice Module, for example MakeCall, PlayFile, SpeakFile, RecordFile, HTTPRequest, etc. Actions usually have parameters.

Each action should have

- attribute: ID [mandatory]. ID must one of the supported actions described below.
- element: par [0..n]. Which represent action parameters. Each par element must have an ID attribute that is supported for this action.

Example 1

- <action id="PlayFile">
- <par id="FileName">d:/voiceclient/menu.vox</par>
- <par id="StopDigits">123*#</par>
- </action>

Example 2

- <action id="SpeakString">
- <par id="String" type="dynamic">state5</par>
- <par id="StopTones">12</par>
- </action>

<par> Element

Each par parameter should have the following attributes:

- **ID** (must, set of parameters ID is listed below for each action)
- **name** (optional)
- **type** (optional)
- **properties** (optional)
- **prepend** (optional)
- **append** (optional)

Example

<par id="FileName" type="dynamic" properties=”trim#|alphanumeric”
prepend=”d:\temp\”append=”.vox”>stateName

</par>

Element <par> Attribute “Type”

Supported attribute types are:

- **fixed** (default)

Example

©HipLink Software 2013. All Rights Reserved.

<par id="String">Welcome to Hiplink Voice Service</par>

- dynamic

If type is dynamic then the element value must have the stateID from the current call. The parameter will take the value of the stateID.

Example

<par id="String" type="dynamic">state5</par>

This means that the value from state5 will be used as the parameter value.

- parameter

Parameters may be passed from Parent Script to Child Script, value of the par element passed as type=parameter cannot be modified (this is close to passing argument by value).

Example

Parent Script set the parameter value:

- <par id="AccessCode" >1234</par>
- Child Script uses the parameter value:
- <par id="String" type="parameter">AccessCode</par>
- variable

Par element can use Script variables to as there value. Variable can be use only after it was declared (in any script)

Example

<par id="3" name="MachineName" type="variable">MachineName</par>

Element <par> Attribute "Properties"

All these properties are applied to the value of the element <par>.

The number of properties may be combined in the "properties" attribute, they should be separated by the '|' symbol.

Supported properties are:

- **cutlastsymbol** - last symbol of the par element value is removed
- **trim***- trims * from the right of the par element value
- **trim#** - trims # from the right of the par element value
- **alphanumeric** – converts numeric value into alphanumeric reversing the following rule:

Each letter in your entry must be converted into a two digit code.

- Letter "A", for example, is on the second button of the phone and is the first of the three letters on this button.
- Dial for letter "A" is 2 1.
- Letter "B" is also on the second button and is the second letter.
- Dial for letter "B" is 2 2.

©HipLink Software 2013. All Rights Reserved.

- For digits, add 0 after the digit you want to enter.

For example, to enter "5" use 5 0. To Enter "7" use 7 0.

For space or any characters other than letters and digits use the star key.

- **Startospace** - character star is replaced by space
- **Startounderscore** - character star is replaced by underscor
- **ttsformat** - some text formatting will be applied (removing extra new lines, etc…) to simplify

TTS conversion (not very much in use).

The following three properties are applied to the par element only if its value is a result of action HTTPrequest.

When HTTPrequest action has been executed by IVR session, the result of it has the following structure:

200 Success

Some Data

Where:

- *result* (default) – response code, in our case result = 200
- *description* - response string, separated by space character from the response code, in our case description = Success
- *data* – some data coming with response, separated by new line from response string, in our case data=Some Data

Example

<par id="UserName" type="dynamic"

properties="trim#|data">stateValidateAccessCode

</par>

Element Par: Attribute Prepend

Attribute Prepend is useful when some extra symbols have to be added in the beginning of to the par value.

Element Par: Attribute Append

Attribute Append is useful when some extra symbols has to be added at the end of to the par value.

Example

Create a file name for the RecordFile action

©HipLink Software 2013. All Rights Reserved.

<par id="FileName" type="dynamic" properties=”trim#|alphanumeric” prepend=”d:\temp\”append=”.vox”>statName

</par>

<transitions> Element

The transitions element is simply a switch statement that contains cases which link the exit events to other states. The last case should be the default case which will catch everything if none of the specific cases matched any of the exit event values.

Each transitions element should have:

- element: case [1..N]. Described below.

Example

<transitions> <case event="1">state5</case>

<case event="2">state3</case>

<case event="MaxDigits">state8</case>

<case event="default">state8</case>

</ transitions >

<case> Element

The case element value (i.e., the string between the tags) must match an ID of a state in this call.

Each case element should have:

- attribute: event**.** The event should either match the exit value of a previous action, or it should be a predefined event
- element value which is an ID of the state to make a transition to

There are three predefined events:

- Event StopDigits happens when any of Stop Digits has been detected by action GetDigits
- Event MaxDigits happens when MaxDigits number has been reached during the action GetDigits
- Event default, assumed to happen when none of another listed transition cases has not been met.

Example

<case event="1">state5</case>

<case event="222">state3</case>

<variables> Element

©HipLink Software 2013. All Rights Reserved.

The variables element is a part of SVXML. It should have one or more elements called var. Each var element should have:

- an attribute ID
- an element value (might be set to some initial value)

Example

<variables> <var id="SecCode">

</var>

<var id="AccessCode">8888</var>

</variables>

Variables are referred by their ID, they can be set by the action SetVariable, and they can be used by any par element, as an element value.

SVXML Actions

Break Method

This method stops execution of current SVXML child script and returns the control back to the calling parent script. It can be called only from child scripts.

Parameters

No parameters.

Result

None

Example

- action id="Break"> </action>

FileDelete Method

This method deletes a file.

Parameter	Description
FileName	Path of the file to be deleted.

Result

- The result is either “@@true” or “@@false”.

Example

<action id="FileDelete">

©HipLink Software 2013. All Rights Reserved.

<par id="FileName" type="parameter" properties="trim#" prepend="d:\tmp\" append=".vox">Pager</par>

</action>

FileExist Method

This method checks if the specified file exists

Parameter	Description
FileName	Path of the file to be checked.

Result

- The result is either “@@true” or “@@false”.

Example

<action id="FileExist">

<par id="FileName" type="parameter" properties="trim#" prepend="d:\tmp\" append=".vox">Pager</par>

</action>

GetDigits Method

Waits for and retrieves touch tone digits on the current Media Resource.

Parameter	Description
MaxDigits	[Optional, Triggers Event]. Maximum number of digits to wait for. The default value is 100.
MaxSeconds	[Optional, Triggers Event]. Maximum seconds to wait. With MaxSeconds set to 0, the method returns immediately. The default value is 6.
MaxSilence	[Optional, Triggers Event]. Maximum seconds to wait between digits. The default value is 3.
StopDigits	[Optional, Triggers Event]. Touch tone digits that will end input immediately. The default value is #.

Result

User input - touch tone digits

©HipLink Software 2013. All Rights Reserved.

Exit Events

- MaxDigits
- MaxSeconds
- MaxSilence
- StopDigits

Example

<action id="GetDigits">

<par id="StopDigits">#</par>

<par id="MaxDigits">20</par>

<par id="MaxSeconds">20</par>

<par id="MaxSilence">5</par>

</action>

GoTo Method

This method starts execution of a SVXML child script.

Parameter	Description
Script	The name of the SVXML child script to execute. If the file name does not include a path, it is considered to be located in the same directory as the parent script or in the directory specified in the templates directory.
State	The ID of the start state in specified SVXML script.
AnyAdtionalParameter	Optional. Any number of additional parameters are supported. Those parameters are considered as incoming parameters for the child script.

Result

- None

Example

<action id="GoTo"> <par id="Script">hl_message_confirmation.svxml</par>

<par id="State">stateHello</par>

<par id="AccessCode" type="dynamic" properties="trim#">stateStart

</par>

<par id="HelloMessage">Hi My Child</par>

</action>

©HipLink Software 2013. All Rights Reserved.

GotoPreviousState Method

- This method returns the control to the previous state inside one script.

Parameters

- No parameters.

Result

- None

Example

```
<state id="stateInvalidEntry" name="Invalid Entry">
<action id="PlayFile">
<par id="FileName">hl_invalid_entry.vox</par>
</action>
<action id="GotoPreviousState">
</action>
</state>
```

HTTPRequest Method

Make a request to the HipLink Server to perform an action. The URL of the server should have been configured in the application interface.

Parameters

Parameters name and number are customizable. Parameters names are required. Parameter's name and value will make a HTTP pair in HTTP request. Parameters should have IDs in numerical form ("1", "2", …) for this action (from 1 to n, where n is the number of parameters).

To send a request like this:

- cmd=RunAct&subcmd=ValidateAccessCode&access_code=3670

An Action element should be created in the following way:

```
<action id="HTTPRequest">
<par id="1" name="cmd">RunAct</par>
<par id="2" name="subcmd">ValidateAccessCode</par>
<par id="3" name="access_code>3670</par>
</action>
```

Example

©HipLink Software 2013. All Rights Reserved.

```
<state id="stateGetReceiverList" name="HTTPPost get receiver list">
<action id="HTTPRequest">
<par id="1" name="cmd">RunAct</par>
<par id="2" name="subcmd">ShowRecv</par>
<par id="3" name="access_code" type="parameter">AccessCode</par>
<par id="4" name="modid">ivr</par>
<par id="5" name="devid">ivr</par>
<par id="6" name="param">recv_pattern</par>
<par id="7" name="value" type="dynamic" properties="alphanumeric">
stateEnterReceiver</par>
</action>
<transitions>
<case event="200">stateChooseReceiver</case>
<case event="default">stateHTTPproblems</case>
</transitions>
</state>
```

Result

The result of this action normally will have three parts:

- Result
- description
- data (always on new line)

Where:

- *result* - a response number
- *description* - a response string, on the same line as the result but separated by a space
- *data* - additional data sent with the response starting on a new line

For example:

- 200 Success receiver John receiverAna natalia

Where:

- *result* = "200" *description* = "Success" *data* = "receiver John receiverAna
- natalia"

©HipLink Software 2013. All Rights Reserved.

When result of HTTPRequest is used for following processing all those three part can be used any time within one call.

To use is a parameter, properties should be specified properly as follows. To play data as a choice list:

<action id="PlayAsChoiceList">

<par id="String" type="dynamic" properties="data">

stateGetReceiverList</par>

</action>

or, in order to play the description of an event: *result description data (always on new line)*

<action id="SpeakString">

<par id="String" type="dynamic" properties="description">

stateGetReceiverList</par>

</action>

MakeCall Method

Places an outbound call on the current Trunk Resource.

Parameter	Description
PhoneNumber	The phone number to dial, including valid characters for pauses, flashes, etc.

Result

One of the following strings: voice, ans_machine, or default.

- voice means that call was picked up by a person,
- ans_machine means that call was picked up by an answering machine,
- default means no connection happened.

Example

<state id="START" name="Making Call">

<action id="MakeCall">

<par id="PhoneNumber">$$phonenumber$$</par>

</action>

<transitions> <case event="voice">stateGreetingPerson</case>

<case event="ans_machine">stateGreetingAnsMachine</case>

©HipLink Software 2013. All Rights Reserved.

<case event="default">stateStop</case>

</transitions>

</state>

PlayAsChoiceList Method

This method plays a list of options, prompts the User to choose one of them, and generates a result based on the User input.

Choices (or Options) are separated by the new line. See example.

Parameter	Description
String	A string representing the options to play, new line should be used as a delimiter between options.
MaxSeconds	Optional. Maximum seconds to wait. With MaxSeconds set to 0, the method returns immediately. The default value is 6.
MaxSilence	Optional. Maximum seconds to wait between digits. The default value is 3.
StopDigits	Touch tone digits that will end input immediately.

Result

The result is the User's input (followed by # key).

Example

<action id="PlayAsChoiceList">

<par id="String">English French German Italian Spanish

</par>

<par id="StopDigits">#</par>

<par id="MaxSeconds">10</par>

<par id="MaxSilence">5</par>

</action>

The following prompt will be played and wait for Users input. Users input are accepted as soon as # keypres:

- For English press 0 and the # key For French press 1 and the # key For German press 2 and the # key For Italian press 3 and the # key For Spanish press 4 and the # key

Any User's input followed by # key is considered as result of this action (for example '1#').

PlayDate Method

©HipLink Software 2013. All Rights Reserved.

Speaks a date, e.g., 'June second nineteen ninety three'.

Parameter	Description
Month	The month to speak, as an integer. For example, use 6 to say “June.”
Day	The day of month to speak (in range 1 to 31).
Year	The year as a 4-digit integer, e.g., 1998 or 2010. To play only the month and day, set Year = 0.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Exit Events

- StopTones

Result

- None

Example

```
<action id="PlayDate">
<par id="Month">10</par>
<par id="Day">23</par>
<par id="Year">2002</par>
<par id="StopTones">01239</par>
</action>
```

PlayDigits Method

Generates the specified DTMF digits on the current Media Resource. Lets you play the DTMF tones without changing the status of the trunk line.

Sometimes, after making a phone call, you need to play DTMF (Dual Tone Multi-Frequency) digits to dial an extension, navigate through a voice menu, or make other DTMF-activated commands. The PlayDigits method lets you play the DTMF tones without changing the status of the trunk line.

Parameter	Description
Digits	List of DTMF digits to play as tones.

©HipLink Software 2013. All Rights Reserved.

Result

- None

Example

```
<action id="PlayDigits">
<par id="Digits">123</par>
</action>
```

PlayDualTone Method

Plays a continuous dual tone on the current Media Resource.

PlayDualTone plays a continuous tone with two sets of amplitudes and frequencies. Most familiar tones from the public telephone network, including touch-tones, are dual tones.

PlayDualTone only generates continuous tones, while PlayDualToneCadence generates cadence tones. A tone with cadence has a regular pattern of periods where the tone is present and where there is silence. In the US, the dial tone is a continuous tone, while the busy tone is a cadence tone.

If a caller types a valid stop tone while a sound file is playing, PlayDualTone stops immediately.

Parameter	Description
Frequency1	Frequency (in Hz) of the first component of the tone. The valid range of frequencies is from 300 to 2000 Hz.
Amplitude1	The amplitude of the first component of the tone, in decibels, ranging from -40 to 2 dB.
Frequency2	Frequency (in Hz) of the first component of the tone. The valid range of frequencies is from 300 to 2000 Hz.
Amplitude2	The amplitude of the first component of the tone, in decibels, ranging from -40 to 2 dB.
Duration	The length of the tone in tenths of a second.
StopTones	[Optional, Triggers Event] Gives a list of touch tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

PlayDualToneCadence Method

©HipLink Software 2013. All Rights Reserved.

Plays a dual cadence tone on the current Media Resource.

PlayDualToneCadence plays a dual cadence tone. A dual tone has two sets of amplitudes and frequencies. A tone with cadence has a regular pattern of periods where the tone is present and where there is silence. In the US, the busy tone is a cadence tone.

Some telephony board and driver combinations do not support cadence tones. Check your API documentation to see if cadence tones are supported.

If a caller types a valid stop tone while a sound file is playing, PlayDualToneCadence stops immediately.

Parameter	Description
Frequency1	Frequency (in Hz) of the first component of the tone. The valid range of frequencies is from 300 to 2000 Hz.
Amplitude1	The amplitude of the first component of the tone, in decibels, ranging from -40 to 2 dB.
Frequency2	Frequency (in Hz) of the first component of the tone. The valid range of frequencies is from 300 to 2000 Hz.
Amplitude2	The amplitude of the first component of the tone, in decibels, ranging from -40 to 2 dB.
Repeat	The number of times to repeat the cadence.
OnTime	Duration of the cadence tone in tenths of a second.
OffTime	Duration of the silence in the cadence, in tenths of a second.
StopTones	[Optional, Triggers Event] Gives a list of touch tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

PlayError Method

Plays a sound file on the current Media Resource, without allowing touch-tone digits to interrupt. This method is typically used for playing error messages that indicate that the caller has misdialed or misunderstood an instruction. Allowing type-ahead or touch-tone digit interruption during an error message may mean that the caller does not hear the error and continues to enter digits believing that there is no problem. PlayError is the equivalent of PlayFile with no stop tones.

Parameter	Description
FileName	The name and path of the file to play (type String).

©HipLink Software 2013. All Rights Reserved.

Result

- None

PlayFile Method

Plays a sound file on the current Media Resource.

This method is typically used for playing error messages that indicate that the caller has mis-dialed or misunderstood an instruction. Allowing type-ahead or touch-tone digit interruption during an error message may mean that the caller does not hear the error and continues to enter digits believing that there is no problem. PlayError is the equivalent of PlayFile with no stop tones.

Parameter	Description
FileName	The name and path of the file to play.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.
SkipSecs	[Optional] The position at which to start playing, in seconds from the start of the file. The default value is 0.
LengthSecs	[Optional] The length of time to play from the file. By default, LenthSecs=0, and PlayFile plays until the end of the file. The default value is 0.

Result

- None

Example

```
<action id="PlayFile">
<par id="FileName">hl_invalid_access_code.vox</par>
</action>
```

PlayFileIfExist Method

Plays a sound file on the current Media Resource if it exists, otherwise no error generated.

Parameter	Description
FileName	The name and path of the file to play.

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.
SkipSecs	[Optional] The position at which to start playing, in seconds from the start of the file. The devault value is 0.
LengthSecs	[Optional] The length of time to play from the file. By default, LengthSecs=0, and PlayFile plays until the end of the file. The default value is 0.

Result

- None.

Example

```
<action id="PlayFileIfExist">
<par id="FileName" type="dynamic"
properties="trim#" append=".vox">stateGetPagerId</par>
</action>
```

PlayFileOfDefault Method

Plays a sound file on the current Media Resource if it exists, otherwise play default file.

Parameter	Description
FileName	The name and path of the file to play.
DefaultFileName	The name and path of the file to play, if FileName file does not exist.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.
SkipSecs	[Optional] The position at which to start playing, in seconds from the start of the file. The devault value is 0.
LengthSecs	[Optional] The length of time to play from the file. By default, LengthSecs=0, and

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
	PlayFile plays until the end of the file. The default value is 0.

Result

- None.

Example

<action id="PlayFileOrDefault">

<par id="FileName" type="dynamic" properties="data" prepend="bmc_status" append=".vox">stateStart</par>

<par id="DefaultFileName">bmc_status0.vox</par>

</action>

PlayInteger Method

Speaks an integer value, e.g. 'twenty three', on the current Media Resource.

Parameter	Description
IntValue	The integer value to speak.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

<action id="PlayInteger">

<par id="IntValue">123</par>

<par id="StopTones">#</par>

</action>

PlayMoney Method

Speaks a money amount, e.g. 'twelve dollars and eight cents'.

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
Dollars	The dollar (or main currency unit, for languages other than North American English) amount to speak.
Cents	The cent (or other subdivision of the currency) amount to speak.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

```
<action id="PlayMoney">
<par id="Dollars">19</par>
<par id="Cents">95</par>
</action>
```

PlayOrdinal Method

Speaks an integer value as 'first', 'second', 'third'...

Parameter	Description
IntValue	The integer value to speak.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

```
<action id="PlayOrdinal">
<par id="IntValue">123</par>
<par id="StopTones">#</par>
```

©HipLink Software 2013. All Rights Reserved.

</action>

PlaySingleTone Method

Plays a continuous single tone on the current Media Resource.

PlaySingleTone plays a tone that has only one amplitude and frequency, without cadence. Most familiar tones from the public telephone network, including touch-tones, are dual tones. To play a continuous dual tone, use PlayDualTone.

PlaySingleTone only generates continuous tones, while PlaySingleToneCadence generates cadence tones. A tone with cadence has a regular pattern of periods where the tone is present and where there is silence. In the US, the dial tone is a continuous tone, while the busy tone is a cadence tone.

If a caller types a valid stop tone while a sound file is playing, PlaySingleTone stops immediately.

Parameter	Description
Frequency	Frequency (Hz) of the tone. The valid range of frequencies is from 300 to 2000 Hz.
Amplitude	The amplitude of the tone, in decibels, ranging from -40 to 2dB.
Duration	The length of the tone, in tenths of a second.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

```
</action><action id="PlaySingleTone">
<par id="Frequency">440</par>
<par id="Amplitude">0</par>
<par id="Duration">20</par>
<par id="StopTones">+</par>
</action>
```

PlayString Method

Speaks a string of characters. For example: 'KS34' is spoken as 'kay ess three four'.

©HipLink Software 2013. All Rights Reserved.

Note: PlayString is not currently supported in Cantonese, Japanese, Korean, Mandarin, or Russian.

Parameter	Description
String	The string of characters to speak.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

<action id="PlayString">

<par id="String" type="dynamic" properties="trim#">stateStart</par>

</action>

PlayTime Method

Speaks a time, e.g. 'nine twenty three a.m.

Parameter	Description
Hour	Hour in range 0 to 23 (using 24-hour clock). Note: If hour parameter value is -1, the current time will be spoken.
Minute	Minutes in range 0 to 59
Second	Seconds in range 0 to 59
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.

Result

- None

Example

<action id="PlayTime">

©HipLink Software 2013. All Rights Reserved.

<par id="Hour">22</par>

<par id="Minute">0</par>

<par id="Second">0</par>

</action>

RecordFile Method

Records sound from the current Media Resource to a sound file.

Parameter	Description
FileName	Name and path of the sound file to record.
MaxSeconds	[Optional, Triggers Event] Maximum time to allow recording, in seconds. The default value is 180.
MaxSilence	[Optional, Triggers Event] Maximum period of silence to allow in recording, measured in seconds. Often a silence time-out is used to end the recording of a new voice mail message, for example. (If this parameter is set to 0, the method ignores silence.) The default value is 0.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the play. Allowed values: a string of zero or more digits, including the star (*) and pound sign (#), or a plus sign (+) to indicate all tones. The default value is 0123456789#*.
Append	[Optional] Determines whether the recording is appended to an existing file of the same name. By default Append is False (0), and the new recording replaces any sound data previously in the file. If Append is True, recording is appended to the end of the current file.

Result

- Either "@@true" or "@@false".

Example

<action id="RecordFile">

<par id="FileName" type="parameter" properties="trim#" prepend="d:\projects\VMSERVICE\records\" append=".vox">Pager</par>

<par id="StopTones">#</par>

</action>

©HipLink Software 2013. All Rights Reserved.

SetResult Method

This action does not do anything, just generate a result.

Parameter	Description
ResultString	Result of this action.

Result

- Result is generated by this action; it is the value of ResultString

Example

```
<state id="stateDefineAvailable" name="Define Pager Availability">
<action id="SetResult">
<par id="ResultString" type="dynamic" properties="data">
stateGetPagerStatus</par>
</action>
<transitions>
<case event="0">stateSendMessage</case>
<case event="1">stateSendMessage</case>
<case event="2">stateSendMessage</case>
<case event="3">statePagerNotAvailable</case>
<case event="default">statePagerNotAvailable</case>
</transitions>
</state>
```

SetVariable Method

Sets the value of previously defined variable

Parameter	Description
VarName	Name of the variable.
VarValue	New value of the variable

Result

- None, just the variable value is changed the action by itself does not generate any result.

©HipLink Software 2013. All Rights Reserved.

Example

<action id="SetVariable">

<par id="VarName">SecCode</par>

<par id="VarValue" type="dynamic" properties="trim#">stateStart</par>

</action>

SpeakFile Method

Speaks text from a file.

Parameter	Description
String	The text string to speak.
Offset	[Optional]
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the speech. The default value is 0123456789#*.

Result

- None.

Example

<action id="SpeakFile">

<par id="FileName">C:\text_messages\welcome.txt</par>

</action>

SpeakString Method

Speaks text from a string.

Parameter	Description
String	The text string to speak.
Offset	[Optional] The position at which to start speaking, in characters from the start of the string. The default value is 0.
StopTones	[Optional, Triggers Event] Gives a list of tones that will interrupt the speech. The default value is 0123456789#*.

©HipLink Software 2013. All Rights Reserved.

Result

- None.

Example

<action id="SpeakString">

<par id="String">blah blah blah</par>

</action>

Stop Method

Disconnects phone line. No parameters. This action is an exception because it does not require a transition state. All calls to Stop will terminate the processing.

Result

- None.

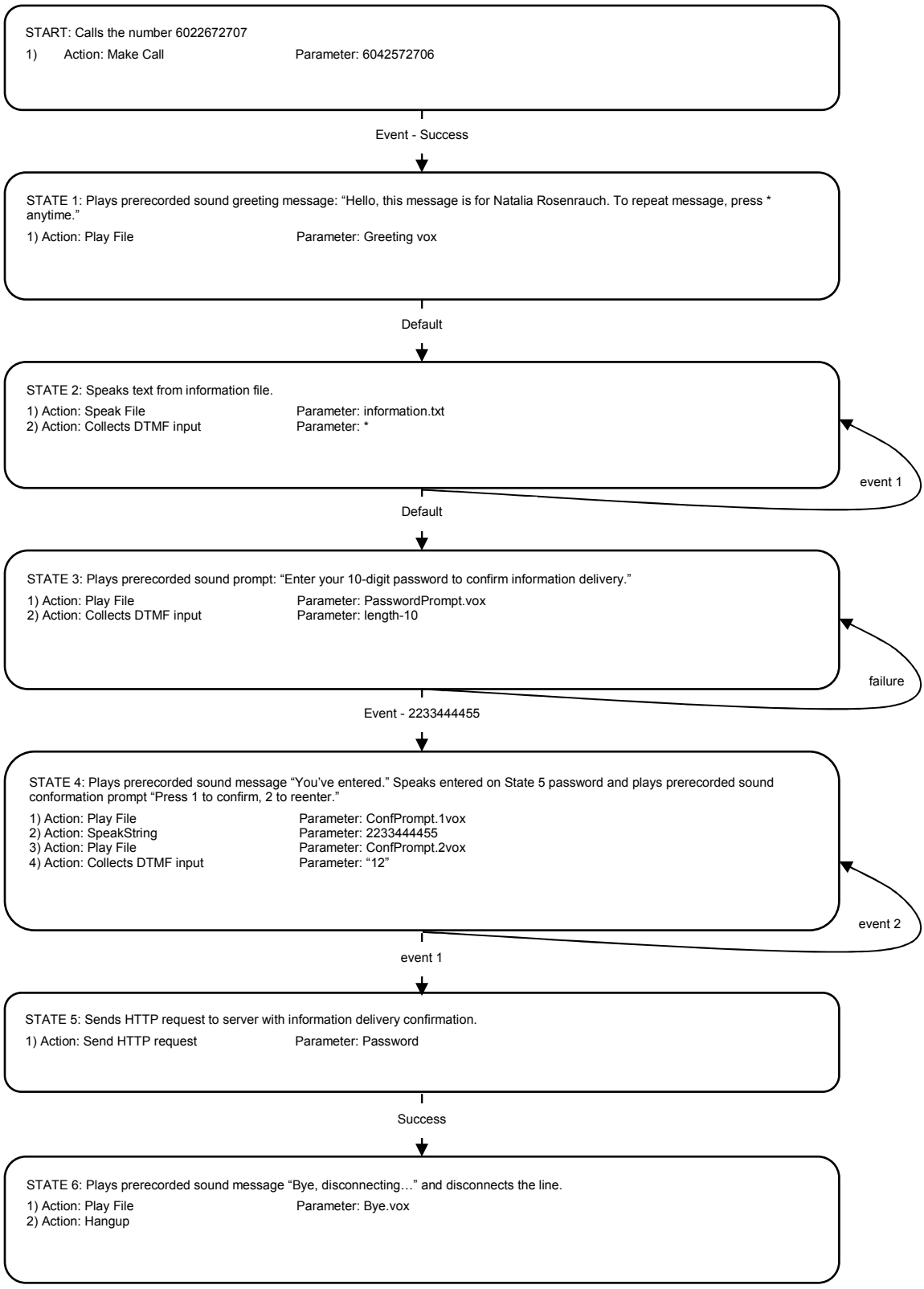
Example IVR Session Implementation

This is an example of an outbound call.

KEY

- Rectangles - states
- Arrows - transitions
- Arrow Subscripts – events

©HipLink Software 2013. All Rights Reserved.



©HipLink Software 2013. All Rights Reserved.

```
<call>
<state id="START" name="Making Call">
<action id="MakeCall">
<par id="PhoneNumber">6042572706</par>
</action>
<transitions>
<case event="Succeed">state2</case>
<case event="default">state7</case>
</transitions>
</state>
<state id="state1" name="Greeting">
<action id="PlayFile">
<par id="FileName">d:/voiceclient/greeting.vox</par>
</action>
<transitions>
<case event="default">state3</case>
</transitions>
</state>
<state id="state2" name="Speaking Information">
<action id="SpeakFile">
<par id="FileName">d:/voiceclient/information.txt</par>
<par id="StopTones">*</par>
</action>
<action id="GetDigits">
<par id="StopDigits">*</par>
</action>
<transitions>
<case event="*">state2</case>
<case event="default">state3</case>
```

©HipLink Software 2013. All Rights Reserved.

```
</transitions>
</state>
<state id="state3" name="Getting Password">
<action id="PlayFile">
<par id="FileName">d:/voiceclient/PasswordPrompt.vox</par>
<par id="StopDigits">#</par>
</action>
<action id="GetDigits">
<par id="StopDigits">#</par>
</action>
<transitions>
<case event="StopDigits">state4</case>
<case event="default">state8</case>
</transitions>
</state>
<state id="state4" name="Confirming Password">
<action id="PlayFile">
<par id="FileName">d:/voiceclient/confprompt1.vox</par>
<par id="StopDigits">12</par>
</action>
<action id="SpeakString">
<par id="String" type="dynamic">state3</par>
<par id="StopTones">12</par>
</action>
<action id="PlayFile">
<par id="FileName">d:/voiceclient/confPrompt2.vox</par>
<par id="StopDigits">12</par>
</action>
<action id="GetDigits">
```
©HipLink Software 2013. All Rights Reserved.

<par id="StopDigits">12</par>

</action>

<transitions>

<case event="1">state5</case>

<case event="2">state3</case>

<case event="default">state8</case>

</transitions>

</state>

<state id="**state5**" name="Sending Delivery Confirmation">

<action id="HTTPRequest">

<par id="1" name= "command">HC</par>

<par id= "2" name="data" type="dynamic">state1</par>

</action>

<transitions >

<case event="Succeed">state6</case>

<case event="default">state8</case>

</transitions>

</state>

<state id="**state6**" name="Disconnecting…">

<action id="PlayFile">

<par id="FileName">d:/voiceclient/Bye.vox</par>

</action>

<action id="Stop">

</action>

</state>

<state id="**state7**" name="Error Playing">

<action id="PlayError">

<par id="FileName">d:/ Error.vox</par>

</action>

©HipLink Software 2013. All Rights Reserved.

<transitions>

<case event="default">state6</case>

</transitions>

</state>

</call>

HipAdm Interface

Introduction

HipLink HipAdm is an executable program that allows administrating HipLink using a command line interface instead of the GUI. HipAdm can be executed either by other processes or by a HipLink User.

Installation

HipAdm can be used either from the same machine where the HipLink server is installed, or from a different machine.

HipAdm comes with the HipLink server installer. Thus, no other installation is required if HipAdm is used on the HipLink server. The executable is located in the HipLink/bin directory (i.e., by default 'C:\Program Files\HipLInk_Software\HipLink\bin' on Windows and '/usr/local/HipLink/bin' on Unix platforms).

If the HipAdm has to be used remotely as a client from a different machine, then it should be copied on that machine in the directory of your choice.

Note: In order to be able to use the HipAdm interface, a designated User of type Non GUI or both has to be set up in the HipLink system. This User specifies the IP of the machine from where the HipAdm will be executed.

Commands

Structure and Syntax

HipAdm, the HipLink Command Line Administrator, is an executable program that sends commands to the HipLink server in order to modify the HipLink database.

HipAdm allows to:

- Add/Delete/Query Receivers
- Add/Delete/Query Groups (i.e., Receiver, On-Duty, Escalation, Rotate, or Follow-Me Group)
- Add/Delete a Receiver to a Group
- Add/Delete a Group to another Group
- Add/Delete/Query Departments

©HipLink Software 2013. All Rights Reserved.

- Add/Delete/Modify/Query Users
- Add/Delete/QueryUser Groups
- Set Department permissions for a User Group
- Set User Group permissions for a User Group.

This section explains the structure and syntax of the HipAdm commands and their parameters.

Authentication

The following authentication parameters are required in all the commands:

Name	Value	Comment
-url	“HipLink server URL”	mandatory, must call the action.exe application, e.g., http://123.123.123.123:8000/cgi-bin/action.exe
-adminusername	“user name”	mandatory, User of type “Non GUI” or “both”
-adminpassword	“user password”	mandatory, User of type “Non GUI” or “both”

Add a Receiver

Command

- Addreceiver

Parameters

Name	Value	Comment
-receivername	“receiver name”	mandatory
-description	“receiver description”	optional
-department	“department name”	mandatory (if Departments’ are enabled by the License Key)
-carrier	“carrier name”	mandatory
-pin	“receiver PIN”	mandatory
-backupcarrier	“alternate carrier”	optional
-backuppin	“alternate PIN”	optional
-receivertype	Alpha | num | 2way	optional, “alpha” by default

Example

- hipadm.exe -addreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe -adminusername admin -adminpassword admin -receivername "John Pager" -pin 1214 -

©HipLink Software 2013. All Rights Reserved.

carrier smtp -department qa -description "hello" -backupcarrier snpp-1 -backuppin 8989 -receivertype alpha

Delete a Receiver

Command

- delreceiver

Parameters

Name	Value	Comment
-receivername	“receiver name”	mandatory

Example

- hipadm.exe -delreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe –adminusername admin -adminpassword admin -receivername "John Pager"

Query Receivers

Command

- queryreceiver

Parameters

Name	Value	Comment
-name	“receiver name”	mandatory (the name can contain “*”)
-department	“department name”	optional (if Departments’ are enabled by the License Key)

Example

- hipadm.exe -queryreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe -name "*"
- -adminusername jude -adminpassword jude

Add a Group

Command

- addgroup

Parameters

Name	Value	Comment
-groupname	“group name”	mandatory

©HipLink Software 2013. All Rights Reserved.

Name	Value	Comment
-description	“group description”	optional
-department	“department name”	mandatory (if Departments’ are enabled by the License Key)
-grouptype	receiver | escalation | onduty | followme | rotate	mandatory

Example

- hipadm.exe -addgroup -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -groupname "Emergency 1" -grouptype escalation -department qa -description "Maintenance Team"

Delete a Group

Command

- Delgroup

Parameters

Name	Value	Comment
-groupname	“group name”	mandatory

Example

- hipadm.exe -delgroup -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -groupname "Emergency 1"

Query Groups

Command

- querygroup

Parameters

Name	Value	Comment
-name	“group name”	mandatory (the name can contain “*”)
-department	“department name”	optional (if Departments’ are enabled by the License Key)

Example

©HipLink Software 2013. All Rights Reserved.

- hipadm.exe -querygroup -name "*" -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -department "m*" -adminusername jude -adminpassword jude

Add a Receiver to a Group

Command

- addrecvtogrp

Parameters

Name	Value	Comment
-member	“receiver name”	mandatory
-groupname	“receiver group name”	mandatory
-delay	“delay”	in minutes, for adding Receivers to Escalation Groups

Example 1

- hipadm.exe -addrecvtogrp -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -groupname "Emergency 1" -member "Jane Mobile"

Example 2

- hipadm.exe -addrecvtogrp -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -groupname "Customer Support" –member "John Doe" -delay 10

Delete a Receiver from a Group

Command

- delrecvfromgrp

Parameters

Name	Value	Comment
-member	“receiver name”	mandatory
-groupname	“receiver group name”	mandatory

Example

- hipadm.exe -delrecvfromgrp -url http://123.123.123.123:8000/cgi-bin/ action.exe

©HipLink Software 2013. All Rights Reserved.

- -adminusername admin -adminpassword admin -groupname "Emergency 1" -member "John Doe"

Add a Group to another Group

Command

- addgrptogrp

Parameters

Name	Value	Comment
-member	“member group name”	mandatory
-groupname	“group name”	mandatory
-delay	“delay”	in minutes, for adding Receivers to Escalation Groups

Example 1

- hipadm.exe -addgrptogrp -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -groupname "Sales Team" -member "Sales Dallas"

Example 2

- hipadm.exe -addgrptogrp -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -groupname "Escalation Support" –member "Level 2" –delay 20

Delete a Group from another Group

Command

- delgrpfromgrp

Parameters

Name	Value	Comment
-member	“member group name”	mandatory
-groupname	“group name”	mandatory

Example

- hipadm.exe -delgrpfromgrp -url http://123.123.123.123:8000/cgi-bin/ action.exe

©HipLink Software 2013. All Rights Reserved.

- -adminusername admin -adminpassword admin -groupname "Sales Team” -member "Sales Dallas"

Add a Department

Command

- adddept

Parameters

Name	Value	Comment
-name	“department name”	mandatory
-description	“department description”	optional

Example

hipadm.exe -adddept -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -name Maintenance -description "Maintenance Department"

Delete a Department

Command

- deldept

Parameters

Name	Value	Comment
-name	“department name”	mandatory

Example

- hipadm.exe -adddept -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -name Maintenance

Query Departments

Command

- querydept

Parameters

Name	Value	Comment
-name	“department name”	Mandatory (the name can contain “*”)

Example

©HipLink Software 2013. All Rights Reserved.

- hipadm.exe -querydept -name "*" -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername jude -adminpassword jude

Add a User

Command

- adduser

Parameters

Name	Value	Comment
-name	“user name”	mandatory
-description	“user description”	optional
-usergroup	“user group”	mandatory
-password	“password”	mandatory
-gui	on	mandatory (for a User of type “GUI” or “both”)
-cli	on	mandatory(for a User of type “Non GUI” or “both)
-email	“e-mail address”	mandatory
-ip	“IP address”	mandatory
-accesscode	“access code”	mandatory

Example

- hipadm.exe -adduser -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -name Jane -description "System Administrator" -usergroup sysAdmin -password 123 -gui on -cli on -email jane@company.com -ip 123.123.123.123
- -accesscode 6789

Modify a User

Command

- moduser

Parameters

Name	Value	Comment
-name	“user name”	mandatory
-newname	“new user name”	mandatory
-description	“user description”	optional

©HipLink Software 2013. All Rights Reserved.

Name	Value	Comment
-usergroup	“user group”	mandatory
-password	“password”	mandatory
-gui	on	mandatory (for a User of type “GUI” or “both”)
-cli	on	mandatory(for a User of type “Non GUI” or “both)
-email	“e-mail address”	mandatory
-ip	“IP address”	mandatory
-accesscode	“access code”	mandatory

Example

- hipadm.exe -adduser -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -name Jane -newname "Jane Brown" -description "System Administrator" -usergroup sysAdmin -password 123 -gui on -cli on -email Jane.Brown@company.com -ip 123.123.123.123 -accesscode 6789

Delete a User

Command

- deluser

Parameters

Name	Value	Comment
-name	“user name”	mandatory

Example

- hipadm.exe -deluser -url http://123.123.123.123:8000/cgi-bin/action.exe -adminusername admin -adminpassword admin -name "Jess Cell"

Query Users

Command

- queryuser

Parameters

Name	Value	Comment
-name	“user name”	Mandatory (the name can contain “*”)

©HipLink Software 2013. All Rights Reserved.

Example

- hipadm.exe -queryuser -name "*" -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername jude -adminpassword jude

Add a User Group

Command

- addusergroup

Name	Value	Comment
-name	“user group name”	mandatory
-description	“user group description”	optional

The following parameters use 'on' to indicate that the permission is granted:

Name	Value	Permission Name	Comment
-carrier	on	add, modify, or delete Carriers	optional
-messenger	on	add, modify, or delete messengers	optional
-grpuser	on	manage Users within this User Group	optional
-grpreport	on	see User Group members report	optional
-monitor	on	modify System Attendant	optional
-egate	on	modify Email Gateway	optional
-dir	on	modify Directories	optional
-system	on	start or stop services	optional
-logs	on	view Logs	optional
-action	on	add, modify, or delete Response Actions	optional
-license	on	modify License Key	optional

The following parameters use 'on' to indicate that the permission is granted:

Name	Value	Permission Name	Comment
-carrier	on	add, modify, or delete Carriers	optional
-messenger	on	add, modify, or delete messengers	optional
-stdsend	on	access Standard Send panel	optional

©HipLink Software 2013. All Rights Reserved.

Name	Value	Permission Name	Comment
-2waysend	on	access Two-Way Send panel	optional
-quicksend	on	access Quick Send panel	optional
-schsend	on	access Schedule Send panel	optional
-cascsend	on	access Custom Escalation Send panel	optional
-faxsend	on	access Fax Send panel	optional
-voicesend	on	access Voice Send panel	optional
-grpuser	on	manage Users within this User Group	optional
-grpreport	on	see User Group members report	optional
-monitor	on	modify System Attendant	optional
egate	on	modify Email Gateway	optional
-dir	on	modify Directories	optional
-system	on	start or stop Services	optional
-logs	on	view Logs	optional
-queue	on	view or modify Queues	optional
-action	on	add, modify, or delete Response Actions	optional
-template	on	add, modify, or delete Templates	optional
-usertemplate	on	use Templates	optional
-license	on	modify License Key	optional

The following parameters are valid only if 'Departments' are not enabled by the **License Key**:

Name	Value	Permission Name	Comment
-sendrece	on	send to individual Receivers	optional
-sendgrp	on	send to Groups	optional
-receiver	on	add, modify, or delete Receivers	optional
-addgroup	on	add or delete Groups	optional
-modgroup	on	modify Groups	optional

©HipLink Software 2013. All Rights Reserved.

Example

Add a new User Group with the name 'operator' that has only ‘template’ and ‘carrier’ permissions:

- hipadm.exe -addusergroup -url http://123.123.123.123:8000/cgi-bin/ action.exe -adminusername admin -adminpassword admin -name operator -description "quality assurance" -carrier on

Delete a User Group

Command

- delusergroup

Parameters

Name	Value	Comment
-name	“user group name”	mandatory

Example

- hipadm.exe -delusergroup -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name operator

Query User Groups

Command

- querygusergroup

Parameters

Name	Value	Comment
-name	“user group name”	Mandatory (the name can contain “*”)

Example

- hipadm.exe -queryusergroup -url http://123.123.123.123:8000/cgi-bin/ action.exe -name "*"
- -adminusername jude -adminpassword jude

Set Department Permissions for a User Group

Command

- setusergroupdept

©HipLink Software 2013. All Rights Reserved.

Parameters

Name	Value	Comment
-name	“user group name”	mandatory
-deptname	“department name”	mandatory

The following parameters use 'on' to indicate that the permission is granted:

Name	Value	Permission Name	Comment
-sendrece	on	Receivers / Send	optional
-modrece	on	Receivers / Manage	optional
-viewrece	on	Receivers / View	optional
-sendgrp	on	Groups / Send	optional
-addgrp	on	Groups / Add / Delete	optional
-modgrp	on	Groups / Modify	optional
-viewgrp	on	Groups / View	optional

Example

- hipadm.exe -setusergroupdept -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name operator -deptname qa -sendrece on -sendgrp on -modgrp on -addgrp on

Set User Group Permissions for a User Group

Command

- setusergroup

Parameters

Name	Value	Comment
-name	“user group name”	mandatory
-ugpname	“user group permissioned”	mandatory

©HipLink Software 2013. All Rights Reserved.

The following parameters use on to indicate that the permission is granted:

Name	Value	Permission Name	Comment
-moduser	on	Manager User Group	optional
-viewrpt	on	View Report	optional

Example

- hipadm.exe -setusergroupugp -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name operator -ugpname tester -moduser on -viewrpt on

Add a Schedule to a Receiver

Command

- addschdtoreceiver

Parameters

Name	Value	Comment
-name	“receiver name”	mandatory
-schdname	“schedule name”	mandatory
-schdtype	“schedule type” [weekly | monthly | nonrecurrent]	mandatory
-starthr	“start hour”	mandatory
-endhr	“end hour”	mandatory
-startmin	“start minute”	optional: default 0
-endmin	“end minute”	optional: default 0

The following parameters are used for recurrent schedules only:

Name	Value	Comment
-startdate	“start date”	optional: default current date. Must be in mm/dd/yyyy format.
-enddate	“end date”	optional: default no end date. Must be in mm/dd/yyyy format.
-recurrences	“number of recurrences”	optional: default 0

©HipLink Software 2013. All Rights Reserved.

The following parameters are used for weekly schedules only:

Name	Value	Comment
-weekdays	“weekdays” [mon. | tue. | wed. | thu. | fri. | sat. | sun.]	mandatory, can be added i.e., mon.tue.fri.
-every	“increment”	optional: default 1

The following parameters are used for monthly schedules only:

Name	Value	Comment
-dayofmonth	“day of the month”	type 1
-weekofmonth	“week of the month”	type 2
-dayofweek	“day of the week”	type 2
-fordays	“last for number of days”	optional: default 1
-every	“increment”	optional: default 1

Example

- hipadm.exe -addschdtoreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name fax -schdname s1 -schdtype weekly -starthr 1 -endhr 6 -weekdays mon.fri.
- hipadm.exe -addschdtoreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name fax -schdname s2 -schdtype weekly -starthr 1 -endhr 6 -weekdays mon.fri. -startdate 3/1/2004 -enddate 3/31/2004
- hipadm.exe -addschdtoreceiver -url http://123.123.123.123:8000/cgi-bin/ action.exe
- -adminusername admin -adminpassword admin -name fax -schdname s3 -schdtype weekly -starthr 1 -endhr 6 -weekdays mon.tue.fri.sat. - recurrences 5 -every 2

Add a Schedule to an On-Duty Group

Command

- addschdtoonduty

Parameters

Name	Value	Comment
-name	“day of the month”	mandatory
-schdname	“week of the month”	mandatory

©HipLink Software 2013. All Rights Reserved.

Name	Value	Comment
-schdtype	“schedule type” [weekly | monthly | nonrecurrent]	mandatory
-membername	“member name”	mandatory
-membertype	“member type” [receiver | group]	mandatory
-starthr	“start hour”	mandatory
-endhr	“end hour”	mandatory
-startmin	“start minute”	optional: default 0
-endmin	“end minute”	optional: default 0

The following parameters are used for recurrent schedules only:

Name	Value	Comment
-startdate	“start date”	optional: default current date. Must be in mm/dd/yyyy format.
-enddate	“end date”	optional: no end date. Must be in mm/dd/yyyy format.
-recurrences	“number of recurrences”	optional: default 0

Note: The enddate and recurrences parameters are mutually exclusive.

The following parameters are used for weekly schedules only:

Name	Value	Comment
-weekdays	“weekdays” [mon. | tue. | wed. | thu. | fri. | sat. | sun.]	mandatory Can be added i.e., mon.tue.fri
-every	“increment”	optional: default 1

The following parameters are used for monthly schedules only:

Name	Value	Comment
-dayofmonth	“day of the month”	type 1

©HipLink Software 2013. All Rights Reserved.

Name	Value	Comment
-weekofmonth	“week of the month”	type 2
-dayofweek	“day of the week”	type 2
-fordays	“last for number of days”	optional: default 1
-every	“increment”	optional: default 1

Example

- hipadm.exe -addschdtoonduty -url http://123.123.123.123:8000/cgi-bin/ action.exe-adminusername admin -adminpassword admin -name duty-1 - membername fax-2 -membertype receiver -schdname s1 -schdtype weekly - starthr 1 -endhr 6 -weekdays mon.tue.fri.sat. -recurrences 5 -every 2
- hipadm.exe -addschdtoonduty -url http://123.123.123.123:8000/cgi-bin/ action.exe-adminusername admin -adminpassword admin -name duty-1 - membername fax-2 -membertype receiver -schdname s2 -schdtype monthly - starthr 5 -endhr 12 -recurrences 5 -every 2 -dayofmonth 15
- hipadm.exe -addschdtoonduty -url http://123.123.123.123:8000/cgi-bin/ action.exe-adminusername admin -adminpassword admin -name duty-1 - membername fax-2 -membertype receiver -schdname s3 -schdtype monthly - starthr 5 -endhr 12 -recurrences 5 -every 2 -weekofmonth 1 -dayofweek 2
- hipadm.exe -addschdtoonduty -url http://123.123.123.123:8000/cgi-bin/ action.exe-adminusername admin -adminpassword admin -name duty-1 - membername rota-2 -membertype group -schdname s4 -schdtype monthly -starthr 18 -endhr 5 -recurrences 5 -fordays 2 -weekofmonth 1 -dayofweek 2

Return Values

Status Codes and Descriptions

When the hipadm.exe is executed, it returns a status code and description, by printing one line of text with 3 or 4 fields separated by white spaces to the standard output.

The return of the CLI has the following format:

- <error/status code> <status text>

The return code 200 means success. All the other returns codes mean failure. Please refer to the CLI Return Values for a complete list of return codes.

©HipLink Software 2013. All Rights Reserved.

SOAP Interface

Introduction

HipLink SOAP interface allows administrating HipLink using a XML Web service instead of the GUI.

HipLink SOAP interface can be run from your browser using the following URL:

- http://<IPaddress>:<PortNumber>/wsdl/wsdlmain.html

HipLink SOAP Web Service Operations

The following operations are supported. For a formal definition, please review the Service Description.

- Add Department
- Delete Department
- Query Department
- Add User
- Modify User
- Delete User
- Query User
- Add User Group
- Delete User Group
- Query User Group
- Set User Group Permissions for Department
- Set User Group Permissions for Other User Group
- Add Receiver
- Delete Receiver
- Query Receiver
- Add Group
- Delete Group
- Query Group
- Add Group Member
- Delete Group Member
- Send Standard Message
- Send Quick Message
- Query Message Status
- Confirm Message
- Query Message History
- Query Group Message Status
- Query Response Action
- Execute Response Action
- Set User Group Response Action Permission

Note: This web service is using http://www.Hiplink.com/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Each XML Web service needs a unique namespace in order for client applications to distinguish it from other services on the Web. http://www.HipLink.com/ is available for XML

©HipLink Software 2013. All Rights Reserved.

Web services that are under development, but published XML Web services should use a more permanent namespace.

Your XML Web service should be identified by a namespace that you control. For example, you can use your company's Internet domain name as part of the namespace. Although many XML Web service namespaces look like URLs, they need not point to actual resources on the Web. (XML Web service namespaces are URIs.)

For XML Web services creating using ASP.NET, the default namespace can be changed using the WebService attribute's Namespace property. The WebService attribute is an attribute applied to the class that contains the XML Web service methods. Below is a code example that sets the namespace to http://microsoft.com/webservices/:

- C#
- [WebService(Namespace="http://microsoft.com/webservices/")]
- public class MyWebService {
- // implementation
- }
- Visual Basic.NET
- <WebService(Namespace:="http://microsoft.com/webservices/")> Public Class
- MyWebService
- ' implementation
- End Class

For more details on XML namespaces, see the W3C recommendation on Namespaces in XML at "http://www.w3.org/TR/REC-xml-names/".

For more details on WSDL, see the WSDL Specification at "http://www.w3.org/TR/wsdl".

For more details on URIs, see RFC 2396 at "http://www.ietf.org/rfc/rfc2396.txt".

Commands

Authentication

The following authentication parameters are required in all the commands:

Name	Type	Value	Comment
username	string	user name e.g., admin	Administrator Name - i.e., a User of type “Non GUI’ or “both” mandatory
password	string	user password e.g., admin	Administrator Password - i.e., a User of type “Non GUI’ or “both” mandatory

Add Department

©HipLink Software 2013. All Rights Reserved.

Parameters

Name	Type	Value	Comment
department_name	string	department name, e.g., Sales Department	Department Name - the name of the department to be added mandatory
description	string	department description e.g, sales department	Department Description - the description of the new department optional
max_receivers	int	Max receivers e.g., 25	Department Receivers – the number of receviers in the department mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddDepartment"


<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddDepartment xmlns="http://www.hiplink.com/">

      <AddDepartmentRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <department_name>string</department_name>
```

©HipLink Software 2013. All Rights Reserved.

```
        <description>string</description>
        <max_receivers>int</max_receivers>
      </AddDepartmentRequest>
    </AddDepartment>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <AddDepartmentResponse xmlns="http://www.hiplink.com/">
      <AddDepartmentResult>
        <response_message>Department &apos;hiplink&apos; is
added</response_message>
        <response_code>200</response_code>
      </AddDepartmentResult>
    </AddDepartmentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddDepartment xmlns="http://www.hiplink.com/">
      <AddDepartmentRequest>
        <credential>
          <username>admin</username>
```

©HipLink Software 2013. All Rights Reserved.

```
            <password>admin</password>
          </credential>
          <department_name>hiplink</department_name>
          <description>test</description>
          <max_receivers>1</max_receivers>
        </AddDepartmentRequest>
      </AddDepartment>
    </soap:Body>
</soap:Envelope>
```

Delete Department

Parameters

Name	Type	Value	Comment
department_name	String	department name, e.g., Sales Department	Department Name - the name of the department to be deleted mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteDepartment"

<?xml version="1.0" encoding="utf-8"?><soap:Envelope
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteDepartment xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
      <DeleteDepartmentRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <department_name>string</department_name>
      </DeleteDepartmentRequest>
    </DeleteDepartment>
  </soap:Body>
</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <DeleteDepartmentResponse xmlns="http://www.hiplink.com/">
      <DeleteDepartmentResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </DeleteDepartmentResult>
    </DeleteDepartmentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

©HipLink Software 2013. All Rights Reserved.

Example

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteDepartment xmlns="http://www.hiplink.com/">

      <DeleteDepartmentRequest>

        <credential>

          <username>admin</username>

          <password>admin</password>

        </credential>

        <department_name>test</department_name>

      </DeleteDepartmentRequest>

    </DeleteDepartment>

  </soap:Body>

</soap:Envelope>
```

Query Department

Parameters

Name	Type	Value	Comment
Name_filter	String	department name e.g., sales, sa*, *	Department Name - the name of the Department to be retrieved, (the name can contain “*”) mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

©HipLink Software 2013. All Rights Reserved.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryDepartment"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryDepartment xmlns="http://www.hiplink.com/">

      <QueryDepartmentRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <name_filter>string</name_filter>

      </QueryDepartmentRequest>

    </QueryDepartment>

  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

    <QueryDepartmentResponse xmlns="http://www.hiplink.com/">

      <QueryDepartmentResult>

        <response_message>string</response_message>

            <response_code>string</response_code>
                <department>

                <string></string>

                 <string></string>
```

©HipLink Software 2013. All Rights Reserved.

```
</department>
    </QueryDepartmentResult>
   </QueryDepartmentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 1

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryDepartment"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryDepartment xmlns="http://www.hiplink.com/">

      <QueryDepartmentRequest>

        <credential>

          <username>admin</username>

          <password>admin</password>

        </credential>

        <name_filter>*</name_filter>

      </QueryDepartmentRequest>

    </QueryDepartment>

  </soap:Body>

</soap:Envelope>
```

Example 2

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
```

©HipLink Software 2013. All Rights Reserved.

```
Content-Length: length
SOAPAction: "QueryDepartment"
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryDepartmentResponse xmlns="http://www.hiplink.com/">
      <QueryDepartmentResult>
        <response_message>Matching department pattern &apos;*&apos;</response_message>
        <response_code>200</response_code>
        <department>
          <string></string>
          <string>AATE</string>
          <string>abc</string>
          <string>ASG</string>
          <string>ATM SERVICING</string>
          <string>AutoFinanceWeb</string>
          <string>Zenta-Spa I and R groups</string>
          <string>zSeries Storage Management</string>
        </department>
      </QueryDepartmentResult>
    </QueryDepartmentResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Add User

Parameters

Name	Type	Value	Comment
name	string	user name e.g., John Doe	User Name mandatory
description	String	user password e.g., jdoe123	User Description **optional**
password	String	user description e.g., Sales Department Manager	User Password **mandatory**

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
user_group	String	user group e.g., sysOper	User Group Name **mandatory**
email_address	String	e-mail address e.g., jdoe@company.com	Email mandatory
access_code	String	access code e.g., 12341234	Access Code optional
enable_cli	Boolean	true|false	CLI **optional** (for a User of type 'Non GUI' or 'both')
enable_gui	Boolean	true|false	GUI optional (for a User of type 'GUI' or 'both')
ip_address	String	IP address e.g., 121.122.123.124	User IP **mandatory** for a User of type 'Non GUI' or 'both'
timezone	Int	User Time Zone	Time Zone **optional**
result_to_email	String	true|false	optional
account_type	string	GUI | Non GUI or both	mandatory
disabled	Boolean	true|false	User Disabled or not optional

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddUser"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddUser xmlns="http://www.hiplink.com/">

      <AddUserRequest>

        <credential>
```

©HipLink Software 2013. All Rights Reserved.

```
      <username>string</username>
      <password>string</password>
    </credential>
    <disabled>bool</disabled>
    <name>string</name>
    <description>string</description>
    <user_group>string</user_group>
    <password>string</password>
    <email_address>string</email_address>
    <result_to_email>bool</result_to_email>
    <access_code>string</access_code>
    <account_type>string</account_type>
    <enable_cli>bool</enable_cli>
    <enable_gui>bool</enable_gui>
    <ip_address>string</ip_address>
    <timezone>string</timezone>
   </AddUserRequest>
  </AddUser>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <AddUserResponse xmlns="http://www.hiplink.com/">
   <AddUserResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
   </AddUserResult>
```

©HipLink Software 2013. All Rights Reserved.

```
  </AddUserResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddUser"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>

  <AddUser xmlns="http://www.hiplink.com/">

   <AddUserRequest>

    <credential>

     <username>admin</username>

     <password>admin</password>

    </credential>

    <disabled>false</disabled>

    <name>owais</name>

    <description>new user</description>

    <user_group>sysadmin</user_group>

    <password>123</password>

    <email_address>t@t.com</email_address>

    <result_to_email>true</result_to_email>

    <access_code>1234</access_code>

    <account_type>HipLink</account_type>

    <enable_cli>true</enable_cli>

    <enable_gui>true</enable_gui>
```

©HipLink Software 2013. All Rights Reserved.

```
      <ip_address>192.168.5.145</ip_address>
      <timezone>Server Time</timezone>
    </AddUserRequest>
  </AddUser>
 </soap:Body>
</soap:Envelope>
```

Modify User

Parameters

Name	Type	Value	Comment
name	string	user name e.g., John Doe	User Name mandatory
description	String	user password e.g., jdoe123	User Description optional
password	String	user description e.g., Sales Department Manager	User Password **mandatory**
user_group	String	user group e.g., sysOper	User Group Name **mandatory**
email_address	String	e-mail address e.g., jdoe@company.com	Email mandatory
access_code	String	access code e.g., 12341234	Access Code optional
enable_cli	Boolean	true|false	CLI optional (for a **User** of type 'Non GUI' or 'both')
enable_gui	Boolean	true|false	GUI optional (for a **User** of type 'GUI' or 'both')
ip_address	String	IP address e.g., 121.122.123.124	User IP **mandatory** for a **User** of type 'Non GUI' or 'both'
timezone	Int	User Time Zone	Time Zone **optional**
result_to_email	String	true|false	optional
account_type	string	GUI|Non GUI or both	mandatory
disabled	Boolean	true|false	User Disabled or not optional

©HipLink Software 2013. All Rights Reserved.

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddUser"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <ModifyUser xmlns="http://www.hiplink.com/">

      <AddUserRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

            <User>

  <disabled>bool</disabled>

        <name>string</name>

        <description>string</description>

        <user_group>string</user_group>

        <password>string</password>

        <email_address>string</email_address>

        <result_to_email>bool</result_to_email>

        <access_code>string</access_code>

        <account_type>string</account_type>

        <enable_cli>bool</enable_cli>

        <enable_gui>bool</enable_gui>
```

©HipLink Software 2013. All Rights Reserved.

```
      <ip_address>string</ip_address>
      <timezone>string</timezone>
          </user>
    </AddUserRequest>
   </ModifyUser>
  </soap:Body>
</soap:Envelope>


HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
   <ModifyUserResponse xmlns="http://www.hiplink.com/">
    <ModifyUserResult>
     <response_message>string</response_message>
     <response_code>string</response_code>
    </ModifyUserResult>
   </ModifyUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<?xml version="1.0" encoding="utf-8"?>
<soap12:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap12="http://www.w3.org/2003/05/soap-envelope">
  <?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
   <ModifyUser xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
    <AddUserRequest>
     <credential>
       <username>admin</username>
       <password>admin</password>
     </credential>
          <User>
        <disabled>false</disabled>
        <name>owais</name>
        <description>test</description>
        <user_group>sysoper</user_group>
        <password>1234</password>
        <email_address>t@t.com</email_address>
        <result_to_email>false</result_to_email>
        <access_code>1234</access_code>
        <account_type>HipLink</account_type>
        <enable_cli>true</enable_cli>
        <enable_gui>true</enable_gui>
        <ip_address>192.168.5.145</ip_address>
        <timezone>Server Time</timezone>
          </User>
    </AddUserRequest>
   </ModifyUser>
  </soap:Body>
</soap:Envelope>
```

Delete User

Parameters

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
user_name	string	user name e.g., John Doe	User Name mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteUser"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteUser xmlns="http://www.hiplink.com/">

      <DeleteUserRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <user_name>string</user_name>

      </DeleteUserRequest>

    </DeleteUser>

  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

    <DeleteUserResponse xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
    <DeleteUserResult>
      <response_message>string</response_message>
      <response_code>string</response_code>
    </DeleteUserResult>
  </DeleteUserResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <soap:Body>
  <DeleteUser xmlns="http://www.hiplink.com/">
   <DeleteUserRequest>
    <credential>
     <username>admin</username>
     <password>admin</password>
    </credential>
    <user_name>owais</user_name>
   </DeleteUserRequest>
  </DeleteUser>
 </soap:Body>
</soap:Envelope>
```

Query User

Parameters

Name	Type	Value	Comment
name_filter	string	user name e.g., John Doe	User Name mandatory (the name can contain “*”)

©HipLink Software 2013. All Rights Reserved.

SOAP Sample

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryUser"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryUser xmlns="http://www.hiplink.com/">

      <QueryUserRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <name_filter>string</name_filter>

      </QueryUserRequest>

    </QueryUser>

  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

    <QueryUserResponse xmlns="http://www.hiplink.com/">

      <QueryUserResult>

        <response_message>string</response_message>
```

©HipLink Software 2013. All Rights Reserved.

```
      <response_code>string</response_code>
      <user>
        <string></string>
        <string></string>
      </user>
    </QueryUserResult>
  </QueryUserResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example 1

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
 <soap:Body>
  <QueryUser xmlns="http://www.hiplink.com/">
   <QueryUserRequest>
    <credential>
     <username>admin</username>
     <password>admin</password>
    </credential>
    <name_filter>*</name_filter>
   </QueryUserRequest>
  </QueryUser>
 </soap:Body>
</soap:Envelope>
```

Add User Group

Parameters

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
user_group_name	string	user group name e.g., Test Group	User Group Name mandatory
description	string	user group description e.g., Test QA Group task	User Group Description **optional**

The following parameters use 'on' to indicate that the permission is granted:

Name	Type	Value	Comment
send_receiver	boolean	True|False	Allow to Send to Receivers - permission to send to individual Receivers **optional**
send_group	boolean	True|False	Allow Send to Groups - permission to send to Groups **optional**
access_system	boolean	True|False	
access_logs	boolean	True|False	Access Log - permission to view Logs **optional**
access_messenger	boolean	True|False	Access Messenger - permission to add, mod-ify, or delete Messengers **optional**
access_template	boolean	True|False	Access Template - permission to add, modify, or delete Templates **optional**
use_template	boolean	True|False	Use Template - permission to use Templates **optional**
access_queues	boolean	True|False	Access Queue - permission to view or modify Queues **optional**
access_license	boolean	True|False	Access License - permission to modify the License Key **optional**
access_directory	boolean	True|False	Access Directory - permission to modify Directories **optional**
add_group	boolean	True|False	Allow or disallow permission to add groups - **optional**
modify_group	boolean	True|False	Allow or disallow permission to modify a group – **optional**
export_group	boolean	True|False	Allow or disallow permission to export a group – **optional**
response_action	boolean	True|False	Allow or disallow permission to execute response actions – **optional**
access_schedule_send	boolean	True|False	Allow Scheduled Send - permission to access the Schedule Send panel **optional**
access_escalation_send	boolean	True|False	Allow Escalation Send - permission to access the Custom Escalation Send panel optional
access_quick_send	boolean	True|False	Allow Quick Send - permission to access the Quick Send panel optional
access_standard_send	boolean	True|False	Allow Standard Send - permission to access the Standard Send panel optional
access_two_way_send	boolean	True|False	Allow Two Way Send - permission to access the Two-Way

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			Send panel optional
access_fax_send	boolean	True|False	Allow Fax Send - permission to access the Fax Send panel **optional**
access_voice_send	boolean	True|False	Allow Voice Send - permission to access the Voice Send panel optional
access_arcims_send	boolean	True|False	Allow ArcICMS Send - permission to access the ArcICMS Send panel **optional**
access_websignup_send	boolean	True|False	Allow Web Sign-up Send - permission to access the Web Sign-up Send panel **optional**
access_quota_send	boolean	True|False	Allow Quota Send - permission to access the Quota Send panel optional
access_attribute_send	boolean	True|False	Allow Attribute Send - permission to access the Attribute Send panel optional
access_email_gateway	boolean	True|False	Access Email Gateway - permission to modify the Email Gateway optional
access_alarm_gateway	boolean	True|False	Access Serial Gateway - permission to modify the Serial Gateway optional
access_snpp_gateway	boolean	True|False	Access Snpp Gateway - permission to modify the Snpp Gateway optional
access_tap_gateway	boolean	True|False	Access Tap Gateway - permission to modify the Tap Gateway optional
access_action	boolean	True|False	Access Action - permission to add, modify, or delete Response Actions optional
access_system_attendant	boolean	True|False	Access System Attendant - permission to modify the System Attendant settings optional
access_user_group	boolean	True|False	Access User Group - permission to manage Users within this User Group optional
access_user_group_report	boolean	True|False	Access Group Report - permission to see User Group members report **optional**
manage_websignup	boolean	True|False	Access web sign-up - permission to manage Web Sign-up profiles optional
manage_websignup_recipient	boolean	True|False	Access web sign-up Recipient - permission to manage Web Sign-up Recipients optional
access_carrier	boolean	True|False	Access Carrier - permission to add, modify, or

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			delete Carriers optional
access_receiver	boolean	True|False	Access Receiver - permission to add, modify, or delete Receivers optional
access_user	boolean	True|False	Mange users within this User group optional
access_email_support	boolean	True|False	Enable Support Email optional

SOAP Sample

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddUserGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddUserGroup xmlns="http://www.hiplink.com/">

      <AddUserGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <access_action>bool</access_action>

        <access_gis_send>bool</access_gis_send>

        <access_attribute_send>bool</access_attribute_send>

        <access_carrier>bool</access_carrier>

        <access_directory>bool</access_directory>
```

©HipLink Software 2013. All Rights Reserved.

<access_email_gateway>**bool**</access_email_gateway>

<access_email_support>**bool**</access_email_support>

<access_escalation_send>**bool**</access_escalation_send>

<access_fax_send>**bool**</access_fax_send>

<access_license>**bool**</access_license>

<access_logs>**bool**</access_logs>

<access_messenger>**bool**</access_messenger>

<access_queues>**bool**</access_queues>

<access_quick_send>**bool**</access_quick_send>

<access_quota_send>**bool**</access_quota_send>

<access_receiver>bool</access_receiver>

<access_schedule_send>**bool**</access_schedule_send>

<access_alarm_gateway>**bool**</access_alarm_gateway>

<access_snpp_gateway>**bool**</access_snpp_gateway>

<access_standard_send>**bool**</access_standard_send>

<access_system>**bool**</access_system>

<access_system_attendant>**bool**</access_system_attendant>

<access_tap_gateway>**bool**</access_tap_gateway>

<access_template>**bool**</access_template>

<access_two_way_send>**bool**</access_two_way_send>

<access_user>**bool**</access_user>

<access_user_group>**bool**</access_user_group>

<access_user_group_report>**bool**</access_user_group_report>

<access_voice_send>**bool**</access_voice_send>

<access_websignup_send>**bool**</access_websignup_send>

<add_group>**bool**</add_group>

<description>**string**</description>

<export_group>**bool**</export_group>

<manage_websignup>**bool**</manage_websignup>

©HipLink Software 2013. All Rights Reserved.

```
      <manage_websignup_recipient>bool</manage_websignup_recipient>
      <modify_group>bool</modify_group>
      <response_action>bool</response_action>
      <send_group>bool</send_group>
      <send_receiver>bool</send_receiver>
      <use_template>bool</use_template>
      <user_group_name>string</user_group_name>
      <access_HNPPermission>bool</access_HNPPermission>
            <set_start_page>int</set_start_page>
    </AddUserGroupRequest>
   </AddUserGroup>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <AddUserGroupResponse xmlns="http://www.hiplink.com/">
      <AddUserGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </AddUserGroupResult>
    </AddUserGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
```

©HipLink Software 2013. All Rights Reserved.

```
Content-Length: length
SOAPAction: "AddUserGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddUserGroup xmlns="http://www.hiplink.com/">

      <AddUserGroupRequest>

        <credential>

          <username>admin</username>

          <password>admin</password>

        </credential>

        <access_action>true</access_action>

        <access_gis_send>true</access_gis_send>

        <access_attribute_send>true</access_attribute_send>

        <access_carrier>true</access_carrier>

        <access_directory>true</access_directory>

        <access_email_gateway>true</access_email_gateway>

        <access_email_support>true</access_email_support>

        <access_escalation_send>true</access_escalation_send>

        <access_fax_send>true</access_fax_send>

        <access_license>true</access_license>

        <access_logs>true</access_logs>

        <access_messenger>true</access_messenger>

        <access_queues>true</access_queues>

        <access_quick_send>true</access_quick_send>

        <access_quota_send>true</access_quota_send>

        <access_receiver>true</access_receiver>

        <access_schedule_send>true</access_schedule_send>
```

©HipLink Software 2013. All Rights Reserved.

```
<access_alarm_gateway>true</access_alarm_gateway>
<access_snpp_gateway>true</access_snpp_gateway>
<access_standard_send>true</access_standard_send>
<access_system>true</access_system>
<access_system_attendant>true</access_system_attendant>
<access_tap_gateway>true</access_tap_gateway>
<access_template>true</access_template>
<access_two_way_send>true</access_two_way_send>
<access_user>true</access_user>
<access_user_group>true</access_user_group>
<access_user_group_report>true</access_user_group_report>
<access_voice_send>true</access_voice_send>
<access_websignup_send>true</access_websignup_send>
<add_group>true</add_group>
<description>test</description>
<export_group>true</export_group>
<manage_websignup>true</manage_websignup>
<manage_websignup_recipient>true</manage_websignup_recipient>
<modify_group>true</modify_group>
<response_action>true</response_action>
<send_group>true</send_group>
<send_receiver>true</send_receiver>
<use_template>true</use_template>
<user_group_name>Test Group</user_group_name>
    <access_HNPPermission>true</access_HNPPermission>
     <set_start_page>2</set_start_page>
  </AddUserGroupRequest>
 </AddUserGroup>
</soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

</soap:Envelope>

Delete User Group

Parameters

Name	Type	Value	Comment
user_group_name	string	user group name e.g., Test Group	User Group Name **mandatory**

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteUserGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteUserGroup xmlns="http://www.hiplink.com/">

      <DeleteUserGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <user_group_name>string</user_group_name>

      </DeleteUserGroupRequest>

    </DeleteUserGroup>

  </soap:Body>

</soap:Envelope>
```

©HipLink Software 2013. All Rights Reserved.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <DeleteUserGroupResponse xmlns="http://www.hiplink.com/">
      <DeleteUserGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </DeleteUserGroupResult>
    </DeleteUserGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteUserGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteUserGroup xmlns="http://www.hiplink.com/">
      <DeleteUserGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <user_group_name>test group</user_group_name>
```

©HipLink Software 2013. All Rights Reserved.

```
    </DeleteUserGroupRequest>
   </DeleteUserGroup>
  </soap:Body>
</soap:Envelope>
```

Query User Group

Parameters

Name	Type	Value	Comment
name_filter	string	user group name e.g., Test Group	User Group Name **mandatory** (the name can contain “*”)

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryUserGroup"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryUserGroup xmlns="http://www.hiplink.com/">

      <QueryUserGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>
```

©HipLink Software 2013. All Rights Reserved.

```
      </credential>
      <name_filter>string</name_filter>
    </QueryUserGroupRequest>
  </QueryUserGroup>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <QueryUserGroupResponse xmlns="http://www.hiplink.com/">
   <QueryUserGroupResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <user_group>
     <string></string>
     <string></string>
    </user_group>
   </QueryUserGroupResult>
  </QueryUserGroupResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example 1

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryUserGroup"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
```

©HipLink Software 2013. All Rights Reserved.

```
<soap:Body>
  <QueryUserGroup xmlns="http://www.hiplink.com/">
    <QueryUserGroupRequest>
      <credential>
        <username>admin</username>
        <password>admin</password>
      </credential>
      <name_filter>*</name_filter>
    </QueryUserGroupRequest>
  </QueryUserGroup>
</soap:Body>
</soap:Envelope>
```

Set User Group Permissions for Department

Parameters

Name	Type	Value	Comment
user_group	string	user group name e.g., Test Grp	User Group Name **mandatory** (the name can contain “*”)
department_name	string	department name e.g., Sales Department	Department name **mandatory**

The following parameters use 'on' to indicate that the permission is granted:

Name	Type	Value	Comment
send_receiver	boolean	true|false	Allow to Send to Receivers from the specified Department optional
manage_receiver	boolean	true|false	Allow to Manage Receivers from the specified Department optional
view_receiver	boolean	true|false	Allow to View Receivers from the specified Department **optional**
add_group	boolean	true|false	Allow to Add/Delete Groups to/from the specified

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			Department **optional**
modify_group	boolean	true|false	Allow to Modify Groups from the specified Department optional
view_group	boolean	true|false	Allow to View Groups from the specified Department **optional**
send_group	boolean	true|false	Allow to Send to Groups from the specified Department **optional**

SOAP Sample

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SetUserGroupDeptPermission"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <SetUserGroupDeptPermission xmlns="http://www.hiplink.com/">

      <SetUserGroupDeptPermRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <group_name>string</group_name>

        <department_name>string</department_name>

        <add_group>bool</add_group>

        <manage_receiver>bool</manage_receiver>

        <modify_group>bool</modify_group>
```

©HipLink Software 2013. All Rights Reserved.

```
      <send_group>bool</send_group>
      <send_receiver>bool</send_receiver>
      <view_group>bool</view_group>
      <view_receiver>bool</view_receiver>
    </SetUserGroupDeptPermRequest>
  </SetUserGroupDeptPermission>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <SetUserGroupDeptPermissionResponse xmlns="http://www.hiplink.com/">
   <SetUserGroupDeptPermissionResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
   </SetUserGroupDeptPermissionResult>
  </SetUserGroupDeptPermissionResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SetUserGroupDeptPermission"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

```
<SetUserGroupDeptPermission xmlns="http://www.hiplink.com/">
 <SetUserGroupDeptPermRequest>
  <credential>
   <username>admin</username>
   <password>admin</password>
  </credential>
  <group_name>usrsend</group_name>
  <department_name>abc</department_name>
  <add_group>true</add_group>
  <manage_receiver>true</manage_receiver>
  <modify_group>true</modify_group>
  <send_group>true</send_group>
  <send_receiver>true</send_receiver>
  <view_group>true</view_group>
  <view_receiver>false</view_receiver>
 </SetUserGroupDeptPermRequest>
</SetUserGroupDeptPermission>
</soap:Body>
</soap:Envelope>
```

Set User Group Permissions for Other User Group

Parameters

Name	Type	Value	Comment
user_group	string	user group name e.g., Test Group	User Group Name **mandatory**
other_user_group	string	user group permissioned e.g., Sales Group	Other User Group Name **mandatory**
modify_user	boolean	true|false	Allow to Manage User - permission to manage the Users

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			belonging to the permissioned User Group **optional**
view_report	boolean	true|false	Allow to View Report - permission to view the reports of messages sent by Users belonging to the permissioned User Group **optional**

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SetUserGroupUgp"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <SetUserGroupUgp xmlns="http://www.hiplink.com/">

      <SetUserGroupUgpRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <other_user_group>string</other_user_group>

        <user_group>string</user_group>

        <modify_user>bool</modify_user>

        <view_report>bool</view_report>

      </SetUserGroupUgpRequest>

    </SetUserGroupUgp>

  </soap:Body>

</soap:Envelope>
```

©HipLink Software 2013. All Rights Reserved.

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <SetUserGroupUgpResponse xmlns="http://www.hiplink.com/">
      <SetUserGroupUgpResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </SetUserGroupUgpResult>
    </SetUserGroupUgpResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SetUserGroupUgp"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetUserGroupUgp xmlns="http://www.hiplink.com/">
      <SetUserGroupUgpRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <other_user_group>usrsend</other_user_group>
```

©HipLink Software 2013. All Rights Reserved.

```
        <user_group>sysoper</user_group>
        <modify_user>true</modify_user>
        <view_report>false</view_report>
      </SetUserGroupUgpRequest>
    </SetUserGroupUgp>
  </soap:Body>
</soap:Envelope>
```

Add Receiver

Parameters

Name	Type	Value	Comment
receiver_name	string	receiver name e.g., Test Rec	Receiver Name **mandatory**
department_name	string	department name e.g., Test Department	Department **mandatory** (only if 'Departments' are enabled by the License Key)
description	string	receiver description e.g., Test Receiver	Description optional
pin	string	receiver PIN e.g., 1231323	PIN mandatory
carrier	string	carrier name e.g., Test	Carrier mandatory
alternate_pin	string	alternate PIN e.g., 134539	Backup PIN optional
alternate_carrier	string	alternate carrier e.g., Other Carrier	Backup Carrier optional
last_name	string	Last Name e.g., Chad	Last Name optional
first_name	string	First Name e.g., Adams	Fist name optional
email_address	string	Email Address e.g., Adams@test.com	Email Address optional
email_failover	boolean	Email failover true|false	Email Failover optional
email_cc	boolean	Email CC true|false	Email CC optional
timezone	int	User Time Zone	Time Zone

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			mandatory
is_alpha	boolean	Alpha charcters true|false	Keep alpha chars optional
device_type	int	Device Type	Device Type mandatory
security_code	string	Security Code	Security Code optional
account_name	string	Account name e.g., admin	Account Name optional
account_password	string	Account Password e.g., admin	Account Password optional
allow_edit	boolean	Allow edit true|false	Edit Receiver optional
allow_schedule	boolean	Allow schedule true|false	update receiver schedule info optional
cover_by_receiver	string	Cover by receiver	Cover By Receiver optional
disabled	boolean	Disabled true|false	Receiver Status optional
enable_voice	boolean	Enable voice true|false	Voice enabled optional
voice_number	string	Voice number e.g., 1234321	Voice Phone Number optional

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddReceiver"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

```
<AddReceiver xmlns="http://www.hiplink.com/">
  <AddReceiverRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <account_name>string</account_name>
    <account_password>string</account_password>
    <allow_edit>bool</allow_edit>
    <allow_schedule>bool</allow_schedule>
    <alternate_carrier />
    <alternate_pin />
    <carrier>string</carrier>
    <cover_by_receiver />
    <department_name>string</department_name>
    <description>string</description>
    <device_type>string</device_type>
    <disabled>bool</disabled>
    <email_address>string</email_address>
    <enable_cc>bool</enable_cc>
    <enable_failover>bool</enable_failover>
    <enable_voice>bool</enable_voice>
    <first_name>bool</first_name>
    <is_alpha>bool</is_alpha>
    <last_name />
    <pin>string</pin>
    <receiver_name>string</receiver_name>
    <security_code>string</security_code>
    <timezone>string</timezone>
```

©HipLink Software 2013. All Rights Reserved.

```
      <voice_number/>
      <assigned_owner>string</assigned_owner>
          <enable_adv_messaging>bool</enable_adv_messaging>
    </AddReceiverRequest>
   </AddReceiver>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
   <AddReceiverResponse xmlns="http://www.hiplink.com/">
    <AddReceiverResult>
     <response_message>string</response_message>
     <response_code>string</response_code>
    </AddReceiverResult>
   </AddReceiverResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddReceiver"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

```
<AddReceiver xmlns="http://www.hiplink.com/">
  <AddReceiverRequest>
    <credential>
      <username>admin</username>
      <password>admin</password>
    </credential>
    <account_name>test</account_name>
    <account_password>test</account_password>
    <allow_edit>true</allow_edit>
    <allow_schedule>true</allow_schedule>
    <alternate_carrier />
    <alternate_pin />
    <carrier>fax</carrier>
    <cover_by_receiver />
    <department_name>abc</department_name>
    <description>test</description>
    <device_type>fax</device_type>
    <disabled>false</disabled>
    <email_address>t@t.com</email_address>
    <enable_cc>false</enable_cc>
    <enable_failover>false</enable_failover>
    <enable_voice>false</enable_voice>
    <first_name>test</first_name>
    <is_alpha>true</is_alpha>
    <last_name />
    <pin>test</pin>
    <receiver_name>umair</receiver_name>
    <security_code>23456</security_code>
    <timezone>Server Time</timezone>
```

©HipLink Software 2013. All Rights Reserved.

```
      <voice_number />
      <assigned_owner>admin</assigned_owner>
  <enable_adv_messaging>true</enable_adv_messaging>
    </AddReceiverRequest>
   </AddReceiver>
  </soap:Body>
</soap:Envelope>
```

Delete Receiver

Parameters

Name	Type	Value	Comment
receiver_name	string	receiver name e.g., Test Receiver	Receiver Name mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteReceiver"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteReceiver xmlns="http://www.hiplink.com/">

      <DeleteReceiverRequest>

        <credential>

          <username>string</username>
```

©HipLink Software 2013. All Rights Reserved.

```
      <password>string</password>
    </credential>
    <receiver_name>string</receiver_name>
   </DeleteReceiverRequest>
  </DeleteReceiver>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <DeleteReceiverResponse xmlns="http://www.hiplink.com/">
   <DeleteReceiverResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
   </DeleteReceiverResult>
  </DeleteReceiverResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteReceiver"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

 <soap:Body>

  <DeleteReceiver xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
    <DeleteReceiverRequest>
      <credential>
        <username>admin</username>
        <password>admin</password>
      </credential>
      <receiver_name>umair</receiver_name>
    </DeleteReceiverRequest>
  </DeleteReceiver>
 </soap:Body>
</soap:Envelope>
```

Query Receiver

Parameters

Name	Type	Value	Comment
name_filter	string	receiver name e.g. Test Receiver	Receiver Name **mandatory** (the name can contain “*”)
dept_filter	string	Department e.g,sales Department	Department Name **mandatory** (the name can contain “*”)

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryReceiver"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryReceiver xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
    <QueryReceiverRequest>
      <credential>
        <username>string</username>
        <password>string</password>
      </credential>
      <name_filter>string</name_filter>
      <dept_filter>string</dept_filter>
    </QueryReceiverRequest>
  </QueryReceiver>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <QueryReceiverResponse xmlns="http://www.hiplink.com/">
   <QueryReceiverResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <receiver>
     <string></string>
     <string></string>
    </receiver>
   </QueryReceiverResult>
  </QueryReceiverResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

©HipLink Software 2013. All Rights Reserved.

Example 1

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryReceiver"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryReceiver xmlns="http://www.hiplink.com/">

      <QueryReceiverRequest>

        <credential>

          <username>admin</username>

          <password>admin</password>

        </credential>

        <name_filter>*</name_filter>

        <dept_filter>abc</dept_filter>

      </QueryReceiverRequest>

    </QueryReceiver>

  </soap:Body>

</soap:Envelope>
```

Add Group

Parameters

Name	Type	Value	Comment
group_name	string	group name e.g., Test Grp	Group Name mandatory
group_type	string	Department e.g., Sales depareenmt	Group Type mandatory
department_name	string	receiver | escalation | onduty | followme | rotate	Department Name **mandatory** (only if 'Depart-ments' are enabled by the License Key)

©HipLink Software 2013. All Rights Reserved.

topic	string	Topic	optional
description	string	group description e.g., Test Description	Description optional
rotating	boolean	true|false	
owner_settings_enable	boolean	Owner Settings true|false	Enable owner settings optional
owner_do_alert	boolean	Alert Owner true|false	Alert Owner of Changes optional
owner_limit_access	boolean	Owner limit true|false	Owner limit optional
owner_user_name	string	Owner User Name e.g., SysAdmin	Owner Name optional

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddGroup xmlns="http://www.hiplink.com/">

      <AddGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <department_name>string</department_name>

        <description>string</description>

        <group_name>string</group_name>
```

©HipLink Software 2013. All Rights Reserved.

```
      <group_type>string</group_type>
      <owner_user_name>string</owner_user_name>
      <topic />
      <owner_do_alert>bool</owner_do_alert>
      <owner_limit_access>bool</owner_limit_access>
      <owner_settings_enable>bool</owner_settings_enable>
      <rotating>bool</rotating>
    </AddGroupRequest>
  </AddGroup>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <AddGroupResponse xmlns="http://www.hiplink.com/">
   <AddGroupResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
   </AddGroupResult>
  </AddGroupResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddGroup"

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

©HipLink Software 2013. All Rights Reserved.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddGroup xmlns="http://www.hiplink.com/">
      <AddGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <department_name>abc</department_name>
        <description>test</description>
        <group_name>new grp</group_name>
        <group_type>broadcast</group_type>
        <owner_user_name>admin</owner_user_name>
        <topic />
        <owner_do_alert>true</owner_do_alert>
        <owner_limit_access>false</owner_limit_access>
        <owner_settings_enable>true</owner_settings_enable>
        <rotating>false</rotating>
      </AddGroupRequest>
    </AddGroup>
  </soap:Body>
</soap:Envelope>
```

Delete Group

Parameters

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
group_name	string	group name e.g., Test Group	Group Name mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteGroup xmlns="http://www.hiplink.com/">

      <DeleteGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <group_name>string</group_name>

      </DeleteGroupRequest>

    </DeleteGroup>

  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
```

©HipLink Software 2013. All Rights Reserved.

```
  <soapenv:Body>
    <DeleteGroupResponse xmlns="http://www.hiplink.com/">
      <DeleteGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </DeleteGroupResult>
    </DeleteGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteGroup"
```

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteGroup xmlns="http://www.hiplink.com/">
      <DeleteGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>new grp</group_name>
      </DeleteGroupRequest>
    </DeleteGroup>
  </soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

</soap:Envelope>

Query Group

Parameters

Name	Type	Value	Comment
name_filter	string	group name e.g., Test Grp	Group Name **mandatory** (the name can contain “*”)
dept_filter	string	Department name e.g., Sales department	department Name **mandatory** (the name can contain “*”)

SOAP Sample

The following is a sample SOAP request and response. The **placeholders** shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryGroup"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <QueryGroup xmlns="http://www.hiplink.com/">

      <QueryGroupRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <name_filter>string</name_filter>

        <dept_filter>string</dept_filter>

      </QueryGroupRequest>

    </QueryGroup>
```

©HipLink Software 2013. All Rights Reserved.

```
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryGroupResponse xmlns="http://www.hiplink.com/">
      <QueryGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
        <group>
          <string></string>
          <string></string>
        </group>
      </QueryGroupResult>
    </QueryGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example 1

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryGroup"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryGroup xmlns="http://www.hiplink.com/">
      <QueryGroupRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
      <credential>
        <username>admin</username>
        <password>admin</password>
      </credential>
      <name_filter>*</name_filter>
      <dept_filter>abc</dept_filter>
    </QueryGroupRequest>
  </QueryGroup>
 </soap:Body>
</soap:Envelope>
```

Add a Group Member

Parameters

Name	Type	Value	Comment
group_name	string	group name e.g., IT Receiver Group	Group Name mandatory
member_name	String	Member name	Member name mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddGroupMember"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <AddGroupMember xmlns="http://www.hiplink.com/">

      <AddGroupMemberRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
      <credential>
        <username>string</username>
        <password>string</password>
      </credential>
      <group_name>string</group_name>
      <member_name>string</member_name>
    </AddGroupMemberRequest>
  </AddGroupMember>
 </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
 <soapenv:Body>
  <AddGroupMemberResponse xmlns="http://www.hiplink.com/">
   <AddGroupMemberResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
   </AddGroupMemberResult>
  </AddGroupMemberResponse>
 </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "AddGroupMember"

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

©HipLink Software 2013. All Rights Reserved.

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <AddGroupMember xmlns="http://www.hiplink.com/">
      <AddGroupMemberRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>edqs</group_name>
        <member_name>abbott, mark</member_name>
      </AddGroupMemberRequest>
    </AddGroupMember>
  </soap:Body>
</soap:Envelope>
```

Delete Group Member

Parameters

Name	Type	Value	Comment
group_name	string	group name e.g. IT Receiver Group	Group Name mandatory
member_name	String	Member name	Member name mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
```

©HipLink Software 2013. All Rights Reserved.

```
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteGroupMember"
```

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

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteGroupMember xmlns="http://www.hiplink.com/">

      <DeleteGroupMemberRequest>

        <credential>

          <username>string</username>

          <password>string</password>

        </credential>

        <group_name>string</group_name>

        <member_name>string</member_name>

      </DeleteGroupMemberRequest>

    </DeleteGroupMember>

  </soap:Body>

</soap:Envelope>
```

```
HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length
```

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

  <soapenv:Body>

    <DeleteGroupMemberResponse xmlns="http://www.hiplink.com/">

      <DeleteGroupMemberResult>

        <response_message>string</response_message>

        <response_code>string</response_code>

      </DeleteGroupMemberResult>

    </DeleteGroupMemberResponse>
```

©HipLink Software 2013. All Rights Reserved.

```
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteGroupMember"

<?xml version="1.0" encoding="utf-8"?>

<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">

  <soap:Body>

    <DeleteGroupMember xmlns="http://www.hiplink.com/">

      <DeleteGroupMemberRequest>

        <credential>

          <username>admin</username>

          <password>admin</password>

        </credential>

        <group_name>edqs</group_name>

        <member_name>abbott, mark</member_name>

      </DeleteGroupMemberRequest>

    </DeleteGroupMember>

  </soap:Body>

</soap:Envelope>
```

Send Standard Message

Parameters

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
recipient_name	string	recipient name e.g., Bill Jones	Recipient Name mandatory
message	String	message body e.g., hello world!	Message - the message body **mandatory**
include_timestamp	boolean	add name and time stamp true| false	Add Name/Time Stamp - defines if the sender name and timestamp should be automatically appended to the message **optional**
include_sender	boolean	Include sender	Include sender optional
hi_priority	boolean	priority	Message priority optional
include_job_id	boolean	Include Job Id	Include Job Id optional
email_receipt	boolean	Email receipt	Email receipt optional
job_id	String	custom Job ID e.g., ID345	Custom Job ID - the Job ID of the message. This Job ID overwrites the one generated by the HipLink server and which is guar-anteed to be unique. If this feature is used, then the User has the responsibility for the Job ID uniqueness **optional**
sender	String	sender name e.g., admin	Sender - specifies the name of the sender of the message. This name is used to replace the CLI (i.e., Non GUI) User in the reports. It can be set to any string because the CLI permissions are based on the Non GUI User settings (i.e., IP address of the machine where the CLI is installed) **optional**
subject	String	Subject	Message Subject optional
delivery_report	String	Delivery report	Message Delivery Report optional

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SendStdMessage"
```

<?xml version="1.0" encoding="utf-8"?>

©HipLink Software 2013. All Rights Reserved.

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendStdMessage xmlns="http://www.hiplink.com/">
      <StandardSendRequest xmlns="">
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <recipient_name>string</recipient_name>
        <message>string</message>
        <include_timestamp>boolean</include_timestamp>
        <include_sender>boolean</include_sender>
        <hi_priority>boolean</hi_priority>
        <include_job_id>boolean</include_job_id>
        <email_receipt>boolean</email_receipt>
        <job_id>string</job_id>
        <sender>string</sender>
        <subject>string</subject>
        <delivery_report>string</delivery_report>
      </StandardSendRequest>
    </SendStdMessage>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

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

©HipLink Software 2013. All Rights Reserved.

```
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendStdMessageResponse xmlns="http://www.hiplink.com/">
      <SendStdMessageResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
        </SendStdMessageResult>
    </SendStdMessageResponse>
  </soap:Body>
</soap:Envelope>
```

Example

```
POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SendStdMessage"

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SendStdMessage xmlns="http://www.hiplink.com/">
      <StandardSendRequest xmlns="">
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <recipient_name>Abbey</recipient_name>
        <message>test message</message>
        <include_timestamp>true</include_timestamp>
        <include_sender>false</include_sender>
        <hi_priority>true</hi_priority>
        <include_job_id>true</include_job_id>
        <email_receipt>true</email_receipt>
        <job_id>1509_1</job_id>
        <sender>test</sender>
        <subject>test message subject</subject>
        <delivery_report>string</delivery_report>
      </StandardSendRequest>
    </SendStdMessage>
  </soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

```
</soap:Envelope>
```

SEND QUICK MESSAGE

Parameters

Name	Type	Value	Comment
include_timestamp	boolean	add name and time stamp true|false	Add Name/Time Stamp - defines if the sender name and timestamp should be automati-cally appended to the message; optional
include_sender	boolean	Include sender true|false	Include sender; optional
hi_priority	boolean	High priority true|false	Message Priority; optional
include_job_id	boolean	Include Job Id true|false	optional
subject	string	Message subject,e.g., Test sub	Message Subject; optional
message	string	message body, e.g., hello world!	Message - the message body; mandatory
authorization_code	string	Authorization code,e.g., 3456	Message Authorization Code; optional
carrier_name	string	carrier name, e.g., Verizon	Carrier - the Carrier used by the Receiver; mandatory
pin	string	recipient PIN, e.g., 1234	PIN - the pin number of the Receiver to which the message will be sent; mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SendQuickMessage"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SendQuickMessage xmlns="http://www.hiplink.com/">
      <SendQuickMessageRequest>
        <credential>
          <username>string</username>
          <password>string</password>
```

©HipLink Software 2013. All Rights Reserved.

```
        </credential>
        <carrier_name>string</carrier_name>
        <hi_priority>bool</hi_priority>
        <include_job_id>bool</include_job_id>
        <include_sender>bool</include_sender>
        <include_timestamp>bool</include_timestamp>
        <message>string</message>
        <pin>string</pin>
        <authorization_code>string</authorization_code>
        <subject>string</subject>
      </SendQuickMessageRequest>
    </SendQuickMessage>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <SendQuickMessageResponse xmlns="http://www.hiplink.com/">
      <SendQuickMessageResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </SendQuickMessageResult>
    </SendQuickMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "SendQuickMessage"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SendQuickMessage xmlns="http://www.hiplink.com/">
      <SendQuickMessageRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <carrier_name>hnp two-way carrier 1</carrier_name>
        <hi_priority>true</hi_priority>
        <include_job_id>true</include_job_id>
        <include_sender>true</include_sender>
        <include_timestamp>true</include_timestamp>
        <message>test message for hnp</message>
        <pin>hnp</pin>
```

©HipLink Software 2013. All Rights Reserved.

```
        <authorization_code>1234</authorization_code>
        <subject>test subject</subject>
      </SendQuickMessageRequest>
    </SendQuickMessage>
  </soap:Body>
</soap:Envelope>
```

QUERY MESSAGE STATUS

Parameters

Name	Type	Value	Comment
job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryMessageStatus"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryMessageStatus xmlns="http://www.hiplink.com/">
      <QueryMessageStatusRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <job_id>string</job_id>
      </QueryMessageStatusRequest>
    </QueryMessageStatus>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryMessageStatusResponse xmlns="http://www.hiplink.com/">
      <QueryMessageStatusResult>
```

©HipLink Software 2013. All Rights Reserved.

```
        <response_message>string</response_message>
        <response_code>string</response_code>
        <status>string</status>
        <failed_reason>string</failed_reason>
      </QueryMessageStatusResult>
    </QueryMessageStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryMessageStatus xmlns="http://www.hiplink.com/">
      <QueryMessageStatusRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <job_id>2031_1</job_id>
      </QueryMessageStatusRequest>
    </QueryMessageStatus>
  </soap:Body>
</soap:Envelope>
```

CONFIRM MESSAGE

Parameters

Name	Type	Value	Comment
job_id	string	Job ID , e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory
comments	string	Comments, e.g., Test Comment	Any comments about the message; optional
custom_status	String	Message Status, e.g., Confirm or Reject	S of the Message as confirm or reject; mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "ConfirmMessage"

©HipLink Software 2013. All Rights Reserved.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ConfirmMessage xmlns="http://www.hiplink.com/">
      <ConfirmMessageRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <job_id>string</job_id>
        <comments>string</comments>
        <custom_status>string</custom_status>
      </ConfirmMessageRequest>
    </ConfirmMessage>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ConfirmMessageResponse xmlns="http://www.hiplink.com/">
      <ConfirmMessageResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </ConfirmMessageResult>
    </ConfirmMessageResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "ConfirmMessage"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ConfirmMessage xmlns="http://www.hiplink.com/">
      <ConfirmMessageRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <job_id>2009_1</job_id>
        <comments>test</comments>
        <custom_status>confirmed</custom_status>
```

©HipLink Software 2013. All Rights Reserved.

```
      </ConfirmMessageRequest>
    </ConfirmMessage>
  </soap:Body>
</soap:Envelope>
```

QUERY MESSAGE HISTORY

Parameters

Name	Type	Value	Comment
job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryMessageHistory"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryMessageHistory xmlns="http://www.hiplink.com/">
      <QueryMessageHistoryRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <job_id>string</job_id>
      </QueryMessageHistoryRequest>
    </QueryMessageHistory>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryMessageHistoryResponse xmlns="http://www.hiplink.com/">
      <QueryMessageHistoryResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
        <message>
          string
```

©HipLink Software 2013. All Rights Reserved.

```
        </message>
        <failed_reason>string</failed_reason>
        <receiver_name>string</receiver_name>
        <department_name>string</department_name>
        <job_history>
          <history_item>
            <timestamp>string</timestamp>
            <pin>string</pin>
            <carrier>string</carrier>
            <status>string</status>
          </history_item>
        </job_history>
      </QueryMessageHistoryResult>
    </QueryMessageHistoryResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryMessageHistory"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryMessageHistory xmlns="http://www.hiplink.com/">
      <QueryMessageHistoryRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <job_id>2009_1</job_id>
      </QueryMessageHistoryRequest>
    </QueryMessageHistory>
  </soap:Body>
</soap:Envelope>
```

QUERY GROUP MESSAGE STATUS

Parameters

Name	Type	Value	Comment
major_job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

SOAP Sample

©HipLink Software 2013. All Rights Reserved.

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryGroupMessageStatus"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryGroupMessageStatus xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <major_job_id>string</major_job_id>
      </QueryGroupMessageStatusRequest>
    </QueryGroupMessageStatus>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryGroupMessageStatusResponse xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
        <message_statuses>
          </message_status>
          <message_status>
            <timestamp>string</timestamp>
            <job_id>string</job_id>
            <receiver_name>string</receiver_name>
            <status>string</status>
            <failed_reason>string</failed_reason>
          </message_status>
        </message_statuses>
      </QueryGroupMessageStatusResult>
    </QueryGroupMessageStatusResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost

©HipLink Software 2013. All Rights Reserved.

Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryGroupMessageStatus"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryGroupMessageStatus xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <major_job_id>2032_1</major_job_id>
      </QueryGroupMessageStatusRequest>
    </QueryGroupMessageStatus>
  </soap:Body>
</soap:Envelope>
```

QUERY RESPONSE ACTION

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: http://www.hiplink.com/QueryResponseAction

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <QueryResponseAction xmlns="http://www.hiplink.com/">
      <QueryResponseActionRequest />
    </QueryResponseAction>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
```

©HipLink Software 2013. All Rights Reserved.

```
    <QueryResponseActionResponse xmlns="http://www.hiplink.com/">
      <QueryResponseActionResult>
        <response_actions>
          <response_action>
            <action_name>string</action_name>
            <action_description>string</action_description>
            <parameters xsi:nil="true" />
          </response_action>
          <response_action>
            <action_name>string</action_name>
            <action_description>string</action_description>
            <parameters xsi:nil="true" />
          </response_action>
        </response_actions>
      </QueryResponseActionResult>
    </QueryResponseActionResponse>
  </soap:Body>
</soap:Envelope>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "QueryResponseAction"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <QueryResponseAction xmlns="http://www.hiplink.com/">
      <QueryResponseActionRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
      </QueryResponseActionRequest>
    </QueryResponseAction>
  </soap:Body>
</soap:Envelope>
```

EXECUTE RESPONSE ACTION

Parameters

Name	Type	Value	Comment
action_name	string	Action name, e.g., Soap	Action Name for which action is being executed; mandatory
Param_name	string	Parameter name, e.g., test	A parameter List with the response action; optional
Param_value	string	Parameter name,	A parameter List with the response action;

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
		e.g., test	optional

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: http://www.hiplink.com/ExecResponseAction

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ExecResponseAction xmlns="http://www.hiplink.com/">
      <ExecResponseActionRequest>
        <action_name>string</action_name>
        <parameters>
          <parameter>
            <param_name>string</param_name>
            <param_value>string</param_value>
          </parameter>
          <parameter>
            <param_name>string</param_name>
            <param_value>string</param_value>
          </parameter>
        </parameters>
      </ExecResponseActionRequest>
    </ExecResponseAction>
  </soap:Body>
</soap:Envelope>
```

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <ExecResponseActionResponse xmlns="http://www.hiplink.com/">
      <ExecResponseActionResult />
    </ExecResponseActionResponse>
  </soap:Body>
</soap:Envelope>
```

Example

©HipLink Software 2013. All Rights Reserved.

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <ExecResponseAction xmlns="http://www.hiplink.com/">
      <ExecResponseActionRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <action_name>Soap</action_name>
        <parameters />
      </ExecResponseActionRequest>
    </ExecResponseAction>
  </soap:Body>
</soap:Envelope>
```

SET USER GROUP RESPONSE ACTION PERMISSION

Parameters

Name	Type	Value	Comment
user_group	string	User group name, e.g., test grp	Group name for which permissions are applied; mandatory
response_action_name	string	Parameter name; e.g., test	Reponse action name; mandatory
Execute_action	boolean	Parameter name; e.g., test	Enable or disable the response action; mandatory

SOAP Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "http://www.hiplink.com/SetUserGroupResponseActionPerm"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetUserGroupResponseActionPerm xmlns="http://www.hiplink.com/">
      <SetUserGroupResponseActionPermRequest>
        <user_group>string</user_group>
        <response_action_name>string</response_action_name>
        <execute_action>boolean</execute_action>
      </SetUserGroupResponseActionPermRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
    </SetUserGroupResponseActionPerm>
  </soap:Body>
</soap:Envelope>

HTTP/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: length

<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema"
xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <SetUserGroupResponseActionPermResponse xmlns="http://www.hiplink.com/">
      <SetUserGroupResponseActionPermResult />
    </SetUserGroupResponseActionPermResponse>
  </soap:Body>
</soap:Envelope>
```

Example

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <SetUserGroupResponseActionPerm xmlns="http://www.hiplink.com/">
      <SetUserGroupResponseActionPermRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <user_group>test grp</user_group>
        <response_action_name>Send</response_action_name>
        <execute_action>true</execute_action>
      </SetUserGroupResponseActionPermRequest>
    </SetUserGroupResponseActionPerm>
  </soap:Body>
</soap:Envelope>
```

WebService Client Operations Snippets

```
public HipLinkServiceRef.AddDepartmentResponse
AddDepartment(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strDepName,
string arg_strDesc, int arg_iNumRecvr)
{
        HipLinkServiceRef.AddDepartmentRequest objAdddepReq = new
        HipLinkServiceRef.AddDepartmentRequest();
        objAdddepReq.credential = arg_credentials;
        objAdddepReq.department_name = arg_strDepName;
        objAdddepReq.description = arg_strDesc;
        objAdddepReq.max_receivers = arg_iNumRecvr;
        return client.AddDepartment(objAdddepReq);
}
```

©HipLink Software 2013. All Rights Reserved.

```
public HipLinkServiceRef.DeleteDepartmentResponse
DeleteDepartment(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strDepName)
{
        HipLinkServiceRef.DeleteDepartmentRequest objDeldepReq = new
        HipLinkServiceRef.DeleteDepartmentRequest();
         objDeldepReq.credential = arg_credentials;
        objDeldepReq.department_name = arg_strDepName;
         return client.DeleteDepartment(objDeldepReq);
}

public HipLinkServiceRef.AddUserResponse AddUser(HipLinkServiceRef.SecurityToken
arg_credentials, string arg_strName, string arg_strDesc, string arg_strUserGroup, string
arg_strPass,string arg_strEmail, bool arg_bIsResulttoEmail, string arg_strAccessCode, bool
arg_bIsDisabled, bool arg_bIscli, bool arg_bIsgui,string arg_strIPaddress)
{
        HipLinkServiceRef.AddUserRequest objAddUserReq = new
        HipLinkServiceRef.AddUserRequest();
        objAddUserReq.credential = arg_credentials;
         objAddUserReq.disabled = arg_bIsDisabled;
        objAddUserReq.name = arg_strName;
        objAddUserReq.description = arg_strDesc;
        objAddUserReq.user_group = arg_strUserGroup;
        objAddUserReq.password = arg_strPass;
        objAddUserReq.email_address = arg_strEmail;
        objAddUserReq.result_to_email = arg_bIsResulttoEmail;
        objAddUserReq.access_code = arg_strAccessCode;
        objAddUserReq.account_type = "HipLink";
        objAddUserReq.enable_cli = arg_bIscli;
        objAddUserReq.enable_gui = arg_bIsgui;
        if (arg_bIscli || (arg_bIscli && arg_bIsgui))
        {
                objAddUserReq.ip_address = arg_strIPaddress;
        }
        objAddUserReq.timezone = "Server Time";
        return client.AddUser(objAddUserReq);
}
```

```
public HipLinkServiceRef.ModifyUserResponse ModifyUser(HipLinkServiceRef.SecurityToken
arg_credentials, string arg_strName, string arg_strDesc, string arg_strUserGroup, string
arg_strPass,string arg_strEmail, bool arg_bIsResulttoEmail, string arg_strAccessCode, bool
arg_bIsDisabled, bool arg_bIscli, bool arg_bIsgui, string arg_strIPaddress)
{
HipLinkServiceRef.AddUserRequest objModifyUserReq = new
HipLinkServiceRef.AddUserRequest();
        objModifyUserReq.credential = arg_credentials;
        objModifyUserReq.disabled = arg_bIsDisabled;
        objModifyUserReq.name = arg_strName;
        objModifyUserReq.description = arg_strDesc;
        objModifyUserReq.user_group = arg_strUserGroup;
```

©HipLink Software 2013. All Rights Reserved.

```
        objModifyUserReq.password = arg_strPass;
        objModifyUserReq.email_address = arg_strEmail;
        objModifyUserReq.result_to_email = arg_bIsResulttoEmail;
        objModifyUserReq.access_code = arg_strAccessCode;
        objModifyUserReq.account_type = "HipLink";
        objModifyUserReq.enable_cli = arg_bIscli;
        objModifyUserReq.enable_gui = arg_bIsgui;
        if (arg_bIscli || (arg_bIscli && arg_bIsgui))
        {
                objModifyUserReq.ip_address = arg_strIPaddress;
        }
        objModifyUserReq.timezone = "Server Time";
        return client.ModifyUser(objModifyUserReq);
}
```

```
public HipLinkServiceRef.DeleteUserResponse DeleteUser(HipLinkServiceRef.SecurityToken
arg_credentials, string arg_strUserName)
{
HipLinkServiceRef.DeleteUserRequest objDelUserReq = new
HipLinkServiceRef.DeleteUserRequest();
        objDelUserReq.credential = arg_credentials;
        objDelUserReq.user_name = arg_strUserName;
        return client.DeleteUser(objDelUserReq);
}
```

```
public HipLinkServiceRef.AddUserGroupResponse
AddUserGroup(HipLinkServiceRef.SecurityToken arg_credentials, bool arg_bIsaccessAction,
bool arg_bIsacessArcicmsSend, bool arg_bIsAccessAttrSend,bool arg_bIsAccessCarrier, bool
arg_bIsaccessDir, bool arg_bIsaccessEmailGateway, bool arg_bIsaccessEmailSup, bool
arg_bIsaccessfaxSend,
bool arg_bIsaccessEscalSend, bool arg_bIsaccessLicns, bool arg_bIsaccessLog, bool
arg_bIsMsngr, bool arg_bIsaccessQueue, bool arg_bIsaccessQuickSnd,bool
arg_bIsaccessQuotaSnd,bool arg_bIsaccessRcvr,bool arg_bIsaccessSchdl, bool
arg_bIsaccessSrlGateWay, bool arg_bIsaccessSnppGateway,bool
arg_bIsaccessStndrdSnd,bool arg_bIsaccessSys,bool arg_bIsaccessSysAttnd, bool
arg_bIsaccessTapGateway,bool arg_bIsaccessTempl,bool arg_bIsaccessTwowaySnd,bool
arg_bIsaccessUsr,bool arg_bIsaccessUsrGrp, bool arg_bIsaccessUsrGrpRpt,bool
arg_bIsaccessVoiceSnd,bool arg_bIsaccessWepSignupsnd,bool  arg_bIsaccessAddGrp,string
arg_strDesc, bool arg_bIsaccessExportGrp, bool arg_bIsManageWebSignup, bool
arg_isManageWebSignupRcpnt, bool arg_bIsModGrp, bool arg_bIsRespActn, bool
arg_bIsSndGrp, bool arg_bIsSendRcvr, bool arg_bIsUseTemplate, string arg_strUserGrpName)
{
HipLinkServiceRef.AddUserGroupRequest objAddUserGrpReq = new
HipLinkServiceRef.AddUserGroupRequest();
         objAddUserGrpReq.credential = arg_credentials;
         objAddUserGrpReq.access_action = arg_bIsaccessAction;
         objAddUserGrpReq.access_gis_send = arg_bIsacessArcicmsSend;
         objAddUserGrpReq.access_attribute_send  = arg_bIsAccessAttrSend;
         objAddUserGrpReq.access_carrier = arg_bIsAccessCarrier;
```

©HipLink Software 2013. All Rights Reserved.

```
        objAddUserGrpReq.access_directory = arg_bIsaccessDir;
        objAddUserGrpReq.access_email_gateway = arg_bIsaccessEmailGateway;
        objAddUserGrpReq.access_email_support = arg_bIsaccessEmailSup;
        objAddUserGrpReq.access_escalation_send = arg_bIsaccessEscalSend;
        objAddUserGrpReq.access_fax_send = arg_bIsaccessfaxSend;
        objAddUserGrpReq.access_license = arg_bIsaccessLicns;
        objAddUserGrpReq.access_logs  = arg_bIsaccessLog;
        objAddUserGrpReq.access_messenger = arg_bIsMsngr;
        objAddUserGrpReq.access_queues = arg_bIsaccessQueue;
        objAddUserGrpReq.access_quick_send = arg_bIsaccessQuickSnd;
        objAddUserGrpReq.access_quota_send = arg_bIsaccessQuotaSnd;
        objAddUserGrpReq.access_receiver = arg_bIsaccessRcvr;
        objAddUserGrpReq.access_schedule_send = arg_bIsaccessSchdl;
        objAddUserGrpReq.access_alarm_gateway = arg_bIsaccessSrlGateWay;
        objAddUserGrpReq.access_snpp_gateway = arg_bIsaccessSnppGateway;
        objAddUserGrpReq.access_standard_send = arg_bIsaccessStndrdSnd;
        objAddUserGrpReq.access_system = arg_bIsaccessSys;
        objAddUserGrpReq.access_system_attendant  = arg_bIsaccessSysAttnd;
        objAddUserGrpReq.access_tap_gateway = arg_bIsaccessTapGateway;
        objAddUserGrpReq.access_template = arg_bIsaccessTempl;
        objAddUserGrpReq.access_two_way_send = arg_bIsaccessTwowaySnd;
        objAddUserGrpReq.access_user = arg_bIsaccessUsr;
        objAddUserGrpReq.access_user_group = arg_bIsaccessUsrGrp;
        objAddUserGrpReq.access_user_group_report = arg_bIsaccessUsrGrpRpt;
        objAddUserGrpReq.access_voice_send = arg_bIsaccessVoiceSnd;
        objAddUserGrpReq.access_websignup_send = arg_bIsaccessWepSignupsnd;
        objAddUserGrpReq.add_group = arg_bIsaccessAddGrp;
        objAddUserGrpReq.description = arg_strDesc;
        objAddUserGrpReq.export_group = arg_bIsaccessExportGrp;
        objAddUserGrpReq.manage_websignup = arg_bIsManageWebSignup;
        objAddUserGrpReq.manage_websignup_recipient =arg_isManageWebSignupRcpnt;
        objAddUserGrpReq.modify_group = arg_bIsModGrp;
        objAddUserGrpReq.response_action = arg_bIsRespActn;
        objAddUserGrpReq.send_group = arg_bIsSndGrp;
        objAddUserGrpReq.send_receiver = arg_bIsSendRcvr;
        objAddUserGrpReq.use_template = arg_bIsUseTemplate;
        objAddUserGrpReq.user_group_name = arg_strUserGrpName.Trim();
        return client.AddUserGroup(objAddUserGrpReq);
}
```

```
public HipLinkServiceRef.DeleteUserGroupResponse
DeleteUserGroup(HipLinkServiceRef.SecurityToken arg_credentials, string
arg_strUsrGrpName)
{
HipLinkServiceRef.DeleteUserGroupRequest objDelUsrGrpReq = new
HipLinkServiceRef.DeleteUserGroupRequest ();
        objDelUsrGrpReq.credential = arg_credentials;
        objDelUsrGrpReq.user_group_name = arg_strUsrGrpName;
        return client.DeleteUserGroup(objDelUsrGrpReq);
}
```

©HipLink Software 2013. All Rights Reserved.

```
public HipLinkServiceRef.SetUserGroupDeptPermissionResponse
SetUserGroupDeptPermission(HipLinkServiceRef.SecurityToken arg_credentials,string
arg_strGrpName, string arg_strDepName, bool arg_isAddGrp, bool arg_IsMngRcvr,
bool arg_IsModifyGrp, bool arg_IsSndGrp, bool arg_IsSndRcvr, bool arg_IsViewGrp, bool
arg_IsViewRcvr)
{
HipLinkServiceRef.SetUserGroupDeptPermRequest objUsrGrpDepPermReq = new
HipLinkServiceRef.SetUserGroupDeptPermRequest();
        objUsrGrpDepPermReq.credential = arg_credentials;
         objUsrGrpDepPermReq.group_name = arg_strGrpName;
         objUsrGrpDepPermReq.department_name = arg_strDepName;
         objUsrGrpDepPermReq.add_group = arg_isAddGrp;
         objUsrGrpDepPermReq.manage_receiver = arg_IsMngRcvr;
         objUsrGrpDepPermReq.modify_group = arg_IsModifyGrp;
         objUsrGrpDepPermReq.send_group = arg_IsSndGrp;
           objUsrGrpDepPermReq.send_receiver = arg_IsSndRcvr;
           objUsrGrpDepPermReq.view_group = arg_IsViewGrp;
            objUsrGrpDepPermReq.view_receiver = arg_IsViewRcvr;
            return client.SetUserGroupDeptPermission(objUsrGrpDepPermReq);
}
```

```
public HipLinkServiceRef.SetUserGroupUgpResponse
SetUserGroupUGP(HipLinkServiceRef.SecurityToken arg_credentials,bool
arg_IsModifyUsr,string arg_strOtherUsrGrp,string arg_StrUsrGrp,bool arg_IsViewRpt)
{
HipLinkServiceRef.SetUserGroupUgpRequest objUsrGrpUGPReq = new
HipLinkServiceRef.SetUserGroupUgpRequest();
        objUsrGrpUGPReq.credential = arg_credentials;
         objUsrGrpUGPReq.modify_user = arg_IsModifyUsr;
        objUsrGrpUGPReq.view_report = arg_IsViewRpt;
         objUsrGrpUGPReq.other_user_group = arg_strOtherUsrGrp;
        objUsrGrpUGPReq.user_group = arg_StrUsrGrp;
        return client.SetUserGroupUgp(objUsrGrpUGPReq);
}
```

```
public HipLinkServiceRef.AddReceiverResponse
AddReceiver(HipLinkServiceRef.SecurityToken arg_credentials, string arg_StrAccName, string
arg_StrAccPass, bool arg_bIsAllowEdt, bool arg_bIsAllowSchdl, string arg_strAltCarrier,string
arg_strAltPIN, string arg_strCarrier, string arg_StrCoverByRcvr, string arg_strDep, string
arg_StrDesc, string arg_strDevType, bool arg_bIsDisabled, string arg_strEmailadd, bool
arg_bIsEmailcc, bool arg_bIsEmailFailOver, bool b_IsEnableVoice, string arg_strFirstName,
bool arg_bIsAlpha, string arg_strLName, string arg_strPIN, string arg_StrRcvrName, string
arg_strSecCode, string arg_strVoiceno,string strAssgnOwner)
{
HipLinkServiceRef.AddReceiverRequest objAddRcvrReq = new
HipLinkServiceRef.AddReceiverRequest();
        bjAddRcvrReq.credential = arg_credentials;
```

©HipLink Software 2013. All Rights Reserved.

```
        objAddRcvrReq.account_name = arg_StrAccName;
         objAddRcvrReq.account_password = arg_StrAccPass;
         objAddRcvrReq.allow_edit = arg_bIsAllowEdt;
         objAddRcvrReq.allow_schedule = arg_bIsAllowSchdl;
         objAddRcvrReq.alternate_carrier = arg_strAltCarrier;
         objAddRcvrReq.alternate_pin = arg_strAltPIN;
         objAddRcvrReq.carrier = arg_strCarrier;
         objAddRcvrReq.cover_by_receiver = arg_StrCoverByRcvr;
         objAddRcvrReq.department_name = arg_strDep;
         objAddRcvrReq.description = arg_StrDesc;
         objAddRcvrReq.device_type = arg_strDevType;
         objAddRcvrReq.disabled = arg_bIsDisabled;
         objAddRcvrReq.email_address = arg_strEmailadd;
         objAddRcvrReq.enable_cc =arg_bIsEmailcc;
         objAddRcvrReq.enable_failover = arg_bIsEmailFailOver;
         objAddRcvrReq.enable_voice = b_IsEnableVoice;
         objAddRcvrReq.first_name = arg_strFirstName;
         objAddRcvrReq.is_alpha = arg_bIsAlpha;
         objAddRcvrReq.last_name = arg_strLName;
         objAddRcvrReq.pin = arg_strPIN;
         objAddRcvrReq.receiver_name = arg_StrRcvrName;
         objAddRcvrReq.security_code = arg_strSecCode;
         objAddRcvrReq.timezone = "Server Time";
         objAddRcvrReq.voice_number = arg_strVoiceno;
         objAddRcvrReq.assigned_owner = strAssgnOwner;
        return client.AddReceiver(objAddRcvrReq);
}
```

```
public HipLinkServiceRef.DeleteReceiverResponse
DeleteReceiver(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strName)
{
HipLinkServiceRef.DeleteReceiverRequest objDelRecReq = new
HipLinkServiceRef.DeleteReceiverRequest();
        objDelRecReq.credential = arg_credentials;
        objDelRecReq.receiver_name = arg_strName;
        return client.DeleteReceiver(objDelRecReq);
}
```

```
public HipLinkServiceRef.AddGroupResponse AddGroup(HipLinkServiceRef.SecurityToken
arg_credentials, string arg_strDepName, string arg_strDesc, string arg_strGrpName, string
arg_strGrpType, bool arg_bIsAlert, bool arg_bIsLimitAccs, bool arg_bIsEnableSettings, string
arg_strUsrName, bool arg_bIsRotating, string arg_strTopic)
{
HipLinkServiceRef.AddGroupRequest objAddGrpReq = new
HipLinkServiceRef.AddGroupRequest();
        objAddGrpReq.credential = arg_credentials;
        objAddGrpReq.department_name = arg_strDepName;
        objAddGrpReq.description = arg_strDesc;
        objAddGrpReq.group_name = arg_strGrpName;
```

©HipLink Software 2013. All Rights Reserved.

```
        objAddGrpReq.group_type = arg_strGrpType;
        objAddGrpReq.owner_do_alert = arg_bIsAlert;
        objAddGrpReq.owner_limit_access = arg_bIsLimitAccs;
        objAddGrpReq.owner_settings_enable = arg_bIsEnableSettings;
        objAddGrpReq.owner_user_name = arg_strUsrName;
        objAddGrpReq.rotating = arg_bIsRotating;
        objAddGrpReq.topic = arg_strTopic;
        return client.AddGroup(objAddGrpReq);
}
```

```
public HipLinkServiceRef.DeleteGroupResponse
DeleteGroup(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strGrpName)
{
HipLinkServiceRef.DeleteGroupRequest objDelGrpReq = new
HipLinkServiceRef.DeleteGroupRequest ();
        objDelGrpReq.credential = arg_credentials;
        objDelGrpReq.group_name = arg_strGrpName;
        return client.DeleteGroup(objDelGrpReq);
}
```

```
public HipLinkServiceRef.AddGroupMemberResponse
AddGroupMember(HipLinkServiceRef.SecurityToken arg_credentials, string
arg_strGrpName,string arg_strMemberName)
{
HipLinkServiceRef.AddGroupMemberRequest objAddGrpMemReq = new
HipLinkServiceRef.AddGroupMemberRequest();
        objAddGrpMemReq.credential = arg_credentials;
        objAddGrpMemReq.group_name = arg_strGrpName;
        objAddGrpMemReq.member_name = arg_strMemberName;
        return client.AddGroupMember(objAddGrpMemReq);
}
```

```
public HipLinkServiceRef.DeleteGroupMemberResponse
DeleteGroupMember(HipLinkServiceRef.SecurityToken arg_credentials, string
arg_strGrpName, string arg_strMemberName)
{
HipLinkServiceRef.DeleteGroupMemberRequest objDelGrpMemReq = new
HipLinkServiceRef.DeleteGroupMemberRequest ();
        objDelGrpMemReq.credential = arg_credentials;
        objDelGrpMemReq.group_name = arg_strGrpName;
        bjDelGrpMemReq.member_name = arg_strMemberName;
        return client.DeleteGroupMember(objDelGrpMemReq);
}
```

```
public HipLinkServiceRef.SendStdMessageResponse
SendStdMessage(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strDelRpt, bool
arg_bIsEmailRecpt, bool arg_bIsHighPrior, bool arg_bIsinclJob_Id, bool arg_bIsInclSender, bool
```

©HipLink Software 2013. All Rights Reserved.

```
arg_bIsinclTime, string arg_strJId, string arg_strMsg, string arg_strRecName, string
arg_strSender, string arg_strSubject)
{
HipLinkServiceRef.StandardSendRequest objSndStdMsgReq = new
HipLinkServiceRef.StandardSendRequest();
        objSndStdMsgReq.credential = arg_credentials;
        objSndStdMsgReq.delivery_report = arg_strDelRpt;
        objSndStdMsgReq.email_receipt = arg_bIsEmailRecpt;
        objSndStdMsgReq.hi_priority = arg_bIsHighPrior;
        objSndStdMsgReq.include_job_id = arg_bIsinclJob_Id;
         include_sender = arg_bIsInclSender;
        objSndStdMsgReq.include_timestamp = arg_bIsinclTime;
        objSndStdMsgReq.job_id = arg_strJId;
        objSndStdMsgReq.message = arg_strMsg;
        objSndStdMsgReq.recipient_name = arg_strRecName;
        objSndStdMsgReq.sender = arg_strSender;
        objSndStdMsgReq.subject = arg_strSubject;
}
```

```
public HipLinkServiceRef.SendQuickMessageResponse
SendQuickMessage(HipLinkServiceRef.SecurityToken arg_credentials, string
arg_StrCarrName, bool arg_bIsHighPrior, bool arg_bIsinclJob_Id, bool arg_bIsInclSender, bool
arg_bIsinclTime, string arg_strMsg, string arg_strPIN, string arg_strAuthCode, string
arg_strSubject)
{
HipLinkServiceRef.SendQuickMessageRequest objSndQuickMsgReq = new
HipLinkServiceRef.SendQuickMessageRequest();
        objSndQuickMsgReq.carrier_name = arg_StrCarrName;
        objSndQuickMsgReq.credential = arg_credentials;
        objSndQuickMsgReq.hi_priority = arg_bIsHighPrior;
        objSndQuickMsgReq.include_job_id = arg_bIsinclJob_Id;
        objSndQuickMsgReq.include_sender = arg_bIsInclSender;
        objSndQuickMsgReq.include_timestamp = arg_bIsinclTime;
        objSndQuickMsgReq.message = arg_strMsg;
        objSndQuickMsgReq.pin = arg_strPIN;
        objSndQuickMsgReq.authorization_code=arg_strAuthCode;
        objSndQuickMsgReq.subject = arg_strSubject;
        return client.SendQuickMessage(objSndQuickMsgReq);
 }
```

```
public HipLinkServiceRef.ConfirmMessageResponse
ConfirmMessage(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strJobId, string
arg_strComments, string arg_strStatus)
{
HipLinkServiceRef.ConfirmMessageRequest objConfirmMsgReq = new
HipLinkServiceRef.ConfirmMessageRequest();
        objConfirmMsgReq.credential = arg_credentials;
        objConfirmMsgReq.job_id = arg_strJobId;
        objConfirmMsgReq.comments = arg_strComments;
```

©HipLink Software 2013. All Rights Reserved.

```
        objConfirmMsgReq.custom_status = arg_strStatus;
         return client.ConfirmMessage(objConfirmMsgReq);
}
```

```
public HipLinkServiceRef.QueryDepartmentResponse
QueryDepartment(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strNamefilter)
{
HipLinkServiceRef.QueryDepartmentRequest objQuerydepReq = new
HipLinkServiceRef.QueryDepartmentRequest();
        objQuerydepReq.credential = arg_credentials;
        objQuerydepReq.name_filter = arg_strNamefilter;
HipLinkServiceRef.QueryDepartmentResponse objRes =
client.QueryDepartment(objQuerydepReq);
         return objRes;
 }
```

```
public HipLinkServiceRef.QueryUserResponse QueryUser(HipLinkServiceRef.SecurityToken
arg_credentials, string arg_strNamefilter)
{
HipLinkServiceRef.QueryUserRequest objQueryUsrReq = new
HipLinkServiceRef.QueryUserRequest();
        objQueryUsrReq.credential = arg_credentials;
        objQueryUsrReq.name_filter = arg_strNamefilter;
        return client.QueryUser(objQueryUsrReq);
}
```

```
public HipLinkServiceRef.QueryUserGroupResponse
QueryUserGroup(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strNamefilter)
{
HipLinkServiceRef.QueryUserGroupRequest objQueryUsrGrpReq = new
HipLinkServiceRef.QueryUserGroupRequest();
        objQueryUsrGrpReq.credential = arg_credentials;
        objQueryUsrGrpReq.name_filter = arg_strNamefilter;
        return client.QueryUserGroup(objQueryUsrGrpReq);
 }
```

```
public HipLinkServiceRef.QueryReceiverResponse
QueryReceiver(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strDepfilter, string
arg_strNamefilter)
{
HipLinkServiceRef.QueryReceiverRequest objQueryRcvrReq = new
HipLinkServiceRef.QueryReceiverRequest();
        objQueryRcvrReq.credential = arg_credentials;
        objQueryRcvrReq.dept_filter = arg_strDepfilter;
        objQueryRcvrReq.name_filter = arg_strNamefilter;
         return client.QueryReceiver(objQueryRcvrReq);
}
```

©HipLink Software 2013. All Rights Reserved.

```
public HipLinkServiceRef.QueryGroupResponse
QueryGroup(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strDepfilter, string
arg_strNamefilter)
{
HipLinkServiceRef.QueryGroupRequest objQueryGrpReq = new
HipLinkServiceRef.QueryGroupRequest();
        objQueryGrpReq.credential = arg_credentials;
        objQueryGrpReq.dept_filter = arg_strDepfilter;
        objQueryGrpReq.name_filter = arg_strNamefilter;
        return client.QueryGroup(objQueryGrpReq);
}
```

```
public HipLinkServiceRef.QueryMessageStatusResponse
QueryMessageStatus(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strJobId)
{
HipLinkServiceRef.QueryMessageStatusRequest objQueryMsgStatusReq = new
HipLinkServiceRef.QueryMessageStatusRequest();
        objQueryMsgStatusReq.credential = arg_credentials;
        objQueryMsgStatusReq.job_id = arg_strJobId;
        return client.QueryMessageStatus(objQueryMsgStatusReq);
}
```

```
public HipLinkServiceRef.QueryMessageHistoryResponse
QueryMessageHistory(HipLinkServiceRef.SecurityToken arg_credentials, string arg_strJobId)
{
HipLinkServiceRef.QueryMessageHistoryRequest objQueryMsgHistoryReq = new
HipLinkServiceRef.QueryMessageHistoryRequest();
       objQueryMsgHistoryReq.credential = arg_credentials;
        objQueryMsgHistoryReq.job_id = arg_strJobId;
         return client.QueryMessageHistory(objQueryMsgHistoryReq);
}
```

```
public HipLinkServiceRef.QueryGroupMessageStatusResponse
QueryGroupMessageStatus(HipLinkServiceRef.SecurityToken arg_credentials, string
arg_strMajorJobId)
{
HipLinkServiceRef.QueryGroupMessageStatusRequest objQueryGrpMsgStatusReq = new
HipLinkServiceRef.QueryGroupMessageStatusRequest();
        objQueryGrpMsgStatusReq.credential = arg_credentials;
        objQueryGrpMsgStatusReq.major_job_id = arg_strMajorJobId;
        return client.QueryGroupMessageStatus(objQueryGrpMsgStatusReq);
}
```

©HipLink Software 2013. All Rights Reserved.

```
public HipLinkServiceRef.SetUserGroupResponseActionPermResponse
SetUserGrpResponseActnPerm(HipLinkServiceRef.SecurityToken SecToken, string
arg_strgrpName, string arg_strRespActnName, bool arg_bisExec)
{
HipLinkServiceRef.SetUserGroupResponseActionPermRequest
objsetUsrGrpRespActReq = new
HipLinkServiceRef.SetUserGroupResponseActionPermRequest();
     objsetUsrGrpRespActReq.credential = SecToken;
     objsetUsrGrpRespActReq.user_group = arg_strgrpName;
     objsetUsrGrpRespActReq.response_action_name = arg_strRespActnName;
     objsetUsrGrpRespActReq.execute_action = arg_bisExec;
     return client.SetUserGroupResponseActionPerm(objsetUsrGrpRespActReq);
 }
```

©HipLink Software 2013. All Rights Reserved.

HipLink REST Interface

Introduction

REST facilitates the transaction between web servers by allowing loose coupling between different services. REST is less strongly typed than its counterpart, SOAP. The REST language uses nouns and verbs, and has an emphasis on readability. Unlike SOAP, REST does not require mandatory XML parsing and does not require a message header to and from a service provider. This ultimately uses less bandwidth. REST error-handling also differs from that used by SOAP.HipLink REST interface allows administrating HipLink using a XML Web service instead of the GUI.

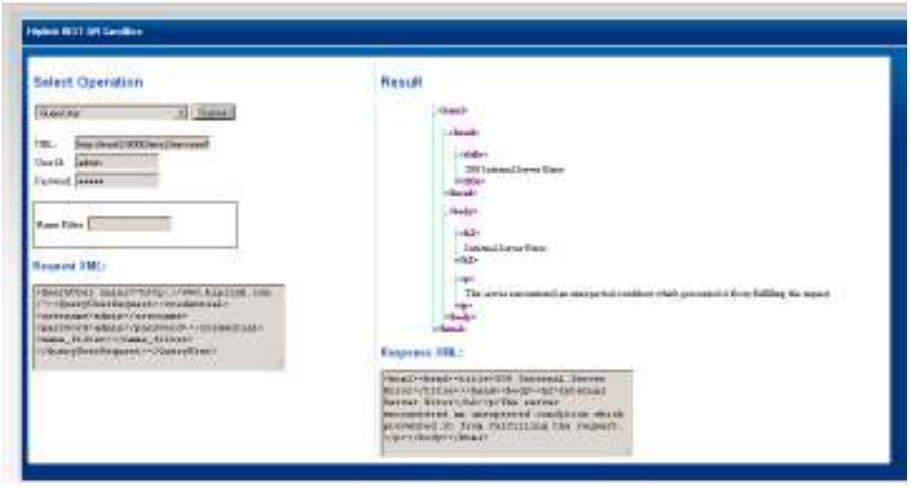
HipLink REST interface can be run from your browser using the following URL:
http://<IPaddress>:<PortNumber>/axis2/services/hiplink/<Entity>/<Operation>

REST Sandbox

Full details are provided below for all service operations but the REST API also includes a Sandbox. The Sandbox is basically a web application that allows you to interact with HipLink using REST API. It is a great learning tool for developers who what to integrate third party apps with HipLink. If HipLink Web Services API Support is enabled in the license key, then you can access sandbox app with URL **http://<Server-IP>:<Port>/rest/index.html** and work as follows:

- Select an operation to perform like Add Department.
- The Sandbox app shows you the target URL for this operation (http://192.168.4.240:8000/axis2/services/hiplink/Department/Add) along with the required parameter input fields like UserID, Password, Department Name, Description, and Maximum Receiver.
- Complete these entries and then press submit.
- The Sandbox API will execute the operation and provide you the results of this REST API Call.
- Moreover it will also display the Request and Response XML Payloads.

This tool makes it very easy for anyone to understand and test our REST API and then use it to integrate with other third party applications.



©HipLink Software 2013. All Rights Reserved.

HipLink REST Web Service Operations

The following operations are supported and can be executed over HTTP using POST method. The request is a REST request and the result will be in simple XML format or JSON.For a formal definition, please review the Service Description.

- Add Department
- Delete Department
- Query Department
- Add User
- Modify User
- Delete User
- Query User
- Add User Group
- Delete User Group
- Query User Group
- Set User Group Permissions for Department
- Set User Group Permissions for Other User Group
- Add Receiver
- Delete Receiver
- Query Receiver
- Add Group
- Delete Group
- Query Group
- Add Group Member
- Delete Group Member
- Send Standard Message
- Send Quick Message
- Query Message Status
- Confirm Message
- Query Message History
- Query Group Message Status
- Set User Group Response Action Permission

Note: This web service is using http://www.hiplink.com/ as its default namespace.

Recommendation: Change the default namespace before the XML Web service is made public.

Commands

Authentication

The following authentication parameters are required in all the commands:

Name	Type	Value	Comment
username	string	user name, e.g., admin	Administrator Name - i.e., a User of type “Non GUI’ or “both”; mandatory
password	string	user password, e.g., admin	Administrator Password - i.e., a User of type “Non GUI’ or “both”; mandatory

©HipLink Software 2013. All Rights Reserved.

ADD DEPARTMENT

Parameters

Name	Type	Value	Comment
department_name	string	department name, e.g., Sales Department	Department Name - the name of the department to be added; mandatory
description	string	department description, e.g, sales department	Department Description - the description of the new department; optional
max_receivers	int	Max receivers, e.g., 25	Department Receivers – the number of receviers in the department; mandatory

REST Sample

The following is a sample SOAP request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Department/Add

Request:

```
<AddDepartment xmlns="http://www.hiplink.com/">
  <AddDepartmentRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <department_name>string</department_name>
    <description>string</description>
    <max_receivers>int</max_receivers>
  </AddDepartmentRequest>
</AddDepartment>
```

Response:

```
<AddDepartmentResponse xmlns="http://www.hiplink.com/">
    <AddDepartmentResult>
      <response_message>Department &apos;hiplink&apos;is
added</response_message>
      <response_code>200</response_code>
    </AddDepartmentResult>
  </AddDepartmentResponse>
```

Example

```
<AddDepartment xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
  <AddDepartmentRequest>
    <credential>
      <username>admin</username>
      <password>admin</password>
    </credential>
    <department_name>hiplink</department_name>
    <description>test</description>
    <max_receivers>1</max_receivers>
  </AddDepartmentRequest>
</AddDepartment>
```

DELETE DEPARTMENT

Parameters

Name	Type	Value	Comment
department_name	String	department name, e.g., Sales Department	Department Name - the name of the department to be deleted; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Department/Delete

Request:

```
<DeleteDepartment xmlns="http://www.hiplink.com/">
  <DeleteDepartmentRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <department_name>string</department_name>
  </DeleteDepartmentRequest>
</DeleteDepartment>
```

Response:

```
<DeleteDepartmentResponse xmlns="http://www.hiplink.com/">
  <DeleteDepartmentResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </DeleteDepartmentResult>
</DeleteDepartmentResponse>
```

Example

```
<DeleteDepartment xmlns="http://www.hiplink.com/">
  <DeleteDepartmentRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
    <credential>
      <username>admin</username>
      <password>admin</password>
    </credential>
    <department_name>test</department_name>
  </DeleteDepartmentRequest>
</DeleteDepartment>
```

QUERY DEPARTMENT

Parameters

Name	Type	Value	Comment
Name_filter	String	department name, e.g., sales, sa*, *	Department Name - the name of the Department to be retrieved, (the name can contain “*”); mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Department/Query

Request:

```
<QueryDepartment xmlns="http://www.hiplink.com/">
  <QueryDepartmentRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <name_filter>string</name_filter>
  </QueryDepartmentRequest>
</QueryDepartment>
```

Response:

```
<QueryDepartmentResponse xmlns="http://www.hiplink.com/">
  <QueryDepartmentResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
     <department>
     <string></string>
     <string></string>
        </department>
  </QueryDepartmentResult>
</QueryDepartmentResponse>
```

Example

```
<QueryDepartment xmlns="http://www.hiplink.com/">
  <QueryDepartmentRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
      <credential>
        <username>admin</username>
        <password>admin</password>
      </credential>
      <name_filter>*</name_filter>
    </QueryDepartmentRequest>
  </QueryDepartment>
```

ADD USER

Parameters

Name	Type	Value	Comment
name	string	user name, e.g., John Doe	User Name; mandatory
description	String	user password, e.g., jdoe123	User Description; optional
password	String	user description, e.g., Sales Department Manager	User Password; mandatory
user_group	String	user group e.g., sysOper	User Group Name; mandatory
email_address	String	e-mail address e.g., jdoe@company.com	Email; mandatory
access_code	String	access code e.g., 12341234	Access Code; optional
enable_cli	Boolean	true|false	CLI; optional (for a User of type 'Non GUI' or 'both')
enable_gui	Boolean	true|false	GUI; optional (for a User of type 'GUI' or 'both')
ip_address	String	IP address e.g., 121.122.123.124	User IP; mandatory for a User of type 'Non GUI' or 'both'
timezone	Int	User Time Zone	Time Zone; optional
result_to_email	String	true|false	optional
account_type	string	GUI | Non GUI or both	mandatory
disabled	Boolean	true|false	User Disabled or not optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/User/Add

Request:

```
<AddUser xmlns="http://www.hiplink.com/">
   <AddUserRequest>
     <credential>
       <username>string</username>
```

©HipLink Software 2013. All Rights Reserved.

```
          <password>string</password>
        </credential>
        <disabled>bool</disabled>
        <name>string</name>
        <description>string</description>
        <user_group>string</user_group>
        <password>string</password>
        <email_address>string</email_address>
        <result_to_email>bool</result_to_email>
        <access_code>string</access_code>
        <account_type>string</account_type>
        <enable_cli>bool</enable_cli>
        <enable_gui>bool</enable_gui>
        <ip_address>string</ip_address>
        <timezone>string</timezone>
      </AddUserRequest>
    </AddUser>
```

Response:

```
    <AddUserResponse xmlns="http://www.hiplink.com/">
      <AddUserResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </AddUserResult>
    </AddUserResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<AddUser xmlns="http://www.hiplink.com/">
      <AddUserRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <disabled>false</disabled>
        <name>owais</name>
        <description>new user</description>
        <user_group>sysadmin</user_group>
        <password>123</password>
        <email_address>t@t.com</email_address>
        <result_to_email>true</result_to_email>
        <access_code>1234</access_code>
        <account_type>HipLink</account_type>
        <enable_cli>true</enable_cli>
        <enable_gui>true</enable_gui>
        <ip_address>192.168.5.145</ip_address>
        <timezone>Server Time</timezone>
      </AddUserRequest>
    </AddUser>
```

MODIFY USER

Parameters

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
name	string	user name, e.g., John Doe	User Name; mandatory
description	String	user password, e.g., jdoe123	User Description; optional
password	String	user description, e.g., Sales Department Manager	User Password; mandatory
user_group	String	user group, e.g., sysOper	User Group Name; mandatory
email_address	String	e-mail address, e.g., jdoe@company.com	Email; mandatory
access_code	String	access code, e.g., 12341234	Access Code; optional
enable_cli	Boolean	true|false	CLI; optional (for a User of type 'Non GUI' or 'both')
enable_gui	Boolean	true|false	GUI; optional (for a User of type 'GUI' or 'both')
ip_address	String	IP address e.g., 121.122.123.124	User IP; mandatory for a User of type 'Non GUI' or 'both'
timezone	Int	User Time Zone	Time Zone; optional
result_to_email	String	true|false	optional
account_type	string	GUI|Non GUI or both	mandatory
disabled	Boolean	true|false	User Disabled or not; optional

REST Sample

The following is a sample REST request and response. The **placeholders** shown need to be replaced with actual values.

URL: /axis2/services/hiplink/User/Modify

Request:

```
<ModifyUser xmlns="http://www.hiplink.com/">
  <AddUserRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
<User>
 <disabled>bool</disabled>
    <name>string</name>
    <description>string</description>
    <user_group>string</user_group>
    <password>string</password>
```

©HipLink Software 2013. All Rights Reserved.

```
        <email_address>string</email_address>
        <result_to_email>bool</result_to_email>
        <access_code>string</access_code>
        <account_type>string</account_type>
        <enable_cli>bool</enable_cli>
        <enable_gui>bool</enable_gui>
        <ip_address>string</ip_address>
        <timezone>string</timezone>
    </user>
      </AddUserRequest>
    </ModifyUser>
```

Response:

```
    <ModifyUserResponse xmlns="http://www.hiplink.com/">
      <ModifyUserResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </ModifyUserResult>
    </ModifyUserResponse>
```

Example

```
<ModifyUser xmlns="http://www.hiplink.com/">
      <AddUserRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <User>
           <disabled>false</disabled>
           <name>owais</name>
           <description>test</description>
           <user_group>sysoper</user_group>
           <password>1234</password>
           <email_address>t@t.com</email_address>
           <result_to_email>false</result_to_email>
           <access_code>1234</access_code>
           <account_type>HipLink</account_type>
           <enable_cli>true</enable_cli>
           <enable_gui>true</enable_gui>
           <ip_address>192.168.5.145</ip_address>
           <timezone>Server Time</timezone>
        </User>
      </AddUserRequest>
    </ModifyUser>
```

DELETE USER

Parameters

Name	Type	Value	Comment
user_name	string	user name, e.g., John Doe	User Name; mandatory

©HipLink Software 2013. All Rights Reserved.

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/User/Delete

Request:

```
    <DeleteUser xmlns="http://www.hiplink.com/">
      <DeleteUserRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <user_name>string</user_name>
      </DeleteUserRequest>
    </DeleteUser>
  </soap:Body>
</soap:Envelope>
```

Response:

```
    <DeleteUserResponse xmlns="http://www.hiplink.com/">
      <DeleteUserResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </DeleteUserResult>
    </DeleteUserResponse>
```

Example

```
<DeleteUser xmlns="http://www.hiplink.com/">
      <DeleteUserRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <user_name>owais</user_name>
      </DeleteUserRequest>
    </DeleteUser>
```

QUERY USER

Parameters

Name	Type	Value	Comment
name_filter	string	user name, e.g., John Doe	User Name; mandatory (the name can contain “*”)

©HipLink Software 2013. All Rights Reserved.

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/User/Query

Request:

```
<QueryUser xmlns="http://www.hiplink.com/">
  <QueryUserRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <name_filter>string</name_filter>
  </QueryUserRequest>
</QueryUser>
```

Response:

```
<QueryUserResponse xmlns="http://www.hiplink.com/">
  <QueryUserResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <user>
      <string></string>
      <string></string>
    </user>
  </QueryUserResult>
</QueryUserResponse>
```

Example

```
<QueryUser xmlns="http://www.hiplink.com/">
      <QueryUserRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <name_filter>*</name_filter>
      </QueryUserRequest>
    </QueryUser>
```

ADD USER GROUP

Parameters

Name	Type	Value	Comment
user_group_name	string	user group name, e.g., Test Group	User Group Name; mandatory
description	string	user group description, e.g., Test QA Group task	User Group Description; optional

©HipLink Software 2013. All Rights Reserved.

The following parameters use 'on' to indicate that the permission is granted:

Name	Type	Value	Comment
send_receiver	boolean	True|False	Allow to Send to Receivers - permission to send to individual Receivers; optional
send_group	boolean	True|False	Allow Send to Groups - permission to send to Groups; optional
access_system	boolean	True|False	
access_logs	boolean	True|False	Access Log - permission to view Logs; optional
access_messenger	boolean	True|False	Access Messenger - permission to add, modify, or delete Messengers; optional
access_template	boolean	True|False	Access Template - permission to add, modify, or delete Templates; optional
use_template	boolean	True|False	Use Template - permission to use Templates; optional
access_queues	boolean	True|False	Access Queue - permission to view or modify Queues; optional
access_license	boolean	True|False	Access License - permission to modify the License Key; optional
access_directory	boolean	True|False	Access Directory - permission to modify Directories; optional
add_group	boolean	True|False	Allow or disallow permission to add groups - **optional**
modify_group	boolean	True|False	Allow or disallow permission to modify a group – **optional**
export_group	boolean	True|False	Allow or disallow permission to export a group – **optional**
response_action	boolean	True|False	Allow or disallow permission to execute response actions – **optional**
access_schedule_send	boolean	True|False	Allow Scheduled Send - permission to access the Schedule Send panel; optional
access_escalation_send	boolean	True|False	Allow Escalation Send - permission to access the Custom Escalation Send panel; optional
access_quick_send	boolean	True|False	Allow Quick Send - permission to access the Quick Send panel; optional
access_standard_send	boolean	True|False	Allow Standard Send - permission to access the Standard Send panel; optional
access_two_way_send	boolean	True|False	Allow Two Way Send - permission to access the Two-Way Send panel; optional
access_fax_send	boolean	True|False	Allow Fax Send - permission to access the Fax Send panel; optional
access_voice_send	boolean	True|False	Allow Voice Send - permission to access the Voice Send panel; optional

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
access_arcims_send	boolean	True|False	Allow ArcICMS Send - permission to access the ArcICMS Send panel; optional
access_websignup_send	boolean	True|False	Allow Web Sign-up Send - permission to access the Web Sign-up Send panel; optional
access_quota_send	boolean	True|False	Allow Quota Send - permission to access the Quota Send panel; optional
access_attribute_send	boolean	True|False	Allow Attribute Send - permission to access the Attribute Send panel; optional
access_email_gateway	boolean	True|False	Access Email Gateway - permission to modify the Email Gateway; optional
access_alarm_gateway	boolean	True|False	Access Serial Gateway - permission to modify the Serial Gateway; optional
access_snpp_gateway	boolean	True|False	Access Snpp Gateway - permission to modify the Snpp Gateway; optional
access_tap_gateway	boolean	True|False	Access Tap Gateway - permission to modify the Tap Gateway; optional
access_action	boolean	True|False	Access Action - permission to add, modify, or delete Response Actions; optional
access_system_attendant	boolean	True|False	Access System Attendant - permission to modify the System Attendant settings; optional
access_user_group	boolean	True|False	Access User Group - permission to manage Users within this User Group; optional
access_user_group_report	boolean	True|False	Access Group Report - permission to see User Group members report; optional
manage_websignup	boolean	True|False	Access web sign-up - permission to manage Web Sign-up profiles; optional
manage_websignup_recipient	boolean	True|False	Access web sign-up Recipient - permission to manage Web Sign-up Recipients; optional
access_carrier	boolean	True|False	Access Carrier - permission to add, modify, or delete Carriers; optional
access_receiver	boolean	True|False	Access Receiver - permission to add, modify, or delete Receivers; optional
access_user	boolean	True|False	Mange users within this User group; optional
access_email_support	boolean	True|False	Enable Support Email; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/UserGroup/Add

Request:

©HipLink Software 2013. All Rights Reserved.

```
<AddUserGroup xmlns="http://www.hiplink.com/">
  <AddUserGroupRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <access_action>bool</access_action>
    <access_gis_send>bool</access_gis_send>
    <access_attribute_send>bool</access_attribute_send>
    <access_carrier>bool</access_carrier>
    <access_directory>bool</access_directory>
    <access_email_gateway>bool</access_email_gateway>
    <access_email_support>bool</access_email_support>
    <access_escalation_send>bool</access_escalation_send>
    <access_fax_send>bool</access_fax_send>
    <access_license>bool</access_license>
    <access_logs>bool</access_logs>
    <access_messenger>bool</access_messenger>
    <access_queues>bool</access_queues>
    <access_quick_send>bool</access_quick_send>
    <access_quota_send>bool</access_quota_send>
    <access_receiver>bool</access_receiver>
    <access_schedule_send>bool</access_schedule_send>
    <access_alarm_gateway>bool</access_alarm_gateway>
    <access_snpp_gateway>bool</access_snpp_gateway>
    <access_standard_send>bool</access_standard_send>
    <access_system>bool</access_system>
    <access_system_attendant>bool</access_system_attendant>
    <access_tap_gateway>bool</access_tap_gateway>
    <access_template>bool</access_template>
    <access_two_way_send>bool</access_two_way_send>
    <access_user>bool</access_user>
    <access_user_group>bool</access_user_group>
    <access_user_group_report>bool</access_user_group_report>
    <access_voice_send>bool</access_voice_send>
    <access_websignup_send>bool</access_websignup_send>
    <add_group>bool</add_group>
    <description>string</description>
    <export_group>bool</export_group>
    <manage_websignup>bool</manage_websignup>
    <manage_websignup_recipient>bool</manage_websignup_recipient>
    <modify_group>bool</modify_group>
    <response_action>bool</response_action>
    <send_group>bool</send_group>
    <send_receiver>bool</send_receiver>
    <use_template>bool</use_template>
    <user_group_name>string</user_group_name>
    <access_HNPPermission>bool</access_HNPPermission>
    <set_start_page>int</set_start_page>
  </AddUserGroupRequest>
</AddUserGroup>
```

Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <AddUserGroupResponse xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
      <AddUserGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </AddUserGroupResult>
    </AddUserGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<AddUserGroup xmlns="http://www.hiplink.com/">
      <AddUserGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <access_action>true</access_action>
        <access_gis_send>true</access_gis_send>
        <access_attribute_send>true</access_attribute_send>
        <access_carrier>true</access_carrier>
        <access_directory>true</access_directory>
        <access_email_gateway>true</access_email_gateway>
        <access_email_support>true</access_email_support>
        <access_escalation_send>true</access_escalation_send>
        <access_fax_send>true</access_fax_send>
        <access_license>true</access_license>
        <access_logs>true</access_logs>
        <access_messenger>true</access_messenger>
        <access_queues>true</access_queues>
        <access_quick_send>true</access_quick_send>
        <access_quota_send>true</access_quota_send>
        <access_receiver>true</access_receiver>
        <access_schedule_send>true</access_schedule_send>
        <access_alarm_gateway>true</access_alarm_gateway>
        <access_snpp_gateway>true</access_snpp_gateway>
        <access_standard_send>true</access_standard_send>
        <access_system>true</access_system>
        <access_system_attendant>true</access_system_attendant>
        <access_tap_gateway>true</access_tap_gateway>
        <access_template>true</access_template>
        <access_two_way_send>true</access_two_way_send>
        <access_user>true</access_user>
        <access_user_group>true</access_user_group>
        <access_user_group_report>true</access_user_group_report>
        <access_voice_send>true</access_voice_send>
        <access_websignup_send>true</access_websignup_send>
        <add_group>true</add_group>
        <description>test</description>
        <export_group>true</export_group>
        <manage_websignup>true</manage_websignup>
        <manage_websignup_recipient>true</manage_websignup_recipient>
        <modify_group>true</modify_group>
        <response_action>true</response_action>
        <send_group>true</send_group>
        <send_receiver>true</send_receiver>
        <use_template>true</use_template>
        <user_group_name>Test Group</user_group_name>
```

©HipLink Software 2013. All Rights Reserved.

```
        <access_HNPPermission>true</access_HNPPermission>
        <set_start_page>2</set_start_page>
      </AddUserGroupRequest>
    </AddUserGroup>
```

DELETE USER GROUP

Parameters

Name	Type	Value	Comment
user_group_name	string	user group name, e.g., Test Group	User Group Name; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/UserGroup/Delete

Request:

```
    <DeleteUserGroup xmlns="http://www.hiplink.com/">
      <DeleteUserGroupRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <user_group_name>string</user_group_name>
      </DeleteUserGroupRequest>
    </DeleteUserGroup>
```

Response:

```
    <DeleteUserGroupResponse xmlns="http://www.hiplink.com/">
      <DeleteUserGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </DeleteUserGroupResult>
    </DeleteUserGroupResponse>
  </soapenv:Body>
</soapenv:Envelope>
```

Example

```
<DeleteUserGroup xmlns="http://www.hiplink.com/">
      <DeleteUserGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <user_group_name>test group</user_group_name>
      </DeleteUserGroupRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
</DeleteUserGroup>
```

QUERY USER GROUP

Parameters

Name	Type	Value	Comment
name_filter	string	user group name, e.g., Test Group	User Group Name; mandatory (the name can contain “*”)

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/UserGroup/Query

Request:

```
<QueryUserGroup xmlns="http://www.hiplink.com/">
  <QueryUserGroupRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <name_filter>string</name_filter>
  </QueryUserGroupRequest>
</QueryUserGroup>
```

Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <QueryUserGroupResponse xmlns="http://www.hiplink.com/">
      <QueryUserGroupResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
        <user_group>
          <string></string>
          <string></string>
        </user_group>
      </QueryUserGroupResult>
    </QueryUserGroupResponse>
```

Example 1

```
<QueryUserGroup xmlns="http://www.hiplink.com/">
      <QueryUserGroupRequest>
        <credential>
```

©HipLink Software 2013. All Rights Reserved.

```
      <username>admin</username>
      <password>admin</password>
    </credential>
    <name_filter>*</name_filter>
  </QueryUserGroupRequest>
</QueryUserGroup>
```

SET USER GROUP PERMISSIONS FOR DEPARTMENT

Parameters

Name	Type	Value	Comment
user_group	string	user group name, e.g., Test Grp	User Group Name; mandatory (the name can contain “*”)
department_name	string	department name, e.g., Sales Department	Department name; mandatory

The following parameters use 'on' to indicate that the permission is granted:

Name	Type	Value	Comment
send_receiver	boolean	true|false	Allow to Send to Receivers from the specified Department; optional
manage_receiver	boolean	true|false	Allow to Manage Receivers from the specified Department; optional
view_receiver	boolean	true|false	Allow to View Receivers from the specified Department; optional
add_group	boolean	true|false	Allow to Add/Delete Groups to/from the specified Department; optional
modify_group	boolean	true|false	Allow to Modify Groups from the specified Department; optional
view_group	boolean	true|false	Allow to View Groups from the specified Department; optional
send_group	boolean	true|false	Allow to Send to Groups from the specified Department; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/UserGroup/DeptPermission/Set

Request:

```
<SetUserGroupDeptPermission xmlns="http://www.hiplink.com/">
  <SetUserGroupDeptPermRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <group_name>string</group_name>
```

©HipLink Software 2013. All Rights Reserved.

```
        <department_name>string</department_name>
        <add_group>bool</add_group>
        <manage_receiver>bool</manage_receiver>
        <modify_group>bool</modify_group>
        <send_group>bool</send_group>
        <send_receiver>bool</send_receiver>
        <view_group>bool</view_group>
        <view_receiver>bool</view_receiver>
      </SetUserGroupDeptPermRequest>
    </SetUserGroupDeptPermission>
```

Response:

```
    <SetUserGroupDeptPermissionResponse xmlns="http://www.hiplink.com/">
      <SetUserGroupDeptPermissionResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </SetUserGroupDeptPermissionResult>
    </SetUserGroupDeptPermissionResponse>
```

Example

```
<SetUserGroupDeptPermission xmlns="http://www.hiplink.com/">
      <SetUserGroupDeptPermRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>usrsend</group_name>
        <department_name>abc</department_name>
        <add_group>true</add_group>
        <manage_receiver>true</manage_receiver>
        <modify_group>true</modify_group>
        <send_group>true</send_group>
        <send_receiver>true</send_receiver>
        <view_group>true</view_group>
        <view_receiver>false</view_receiver>
      </SetUserGroupDeptPermRequest>
    </SetUserGroupDeptPermission>
```

SET USER GROUP PERMISSIONS FOR OTHER USER GROUP

Parameters

Name	Type	Value	Comment
user_group	string	user group name e.g., Test Group	User Group Name; mandatory
other_user_group	string	user group permissioned e.g., Sales Group	Other User Group Name; mandatory
modify_user	boolean	true|false	Allow to Manage User - permission to manage the Users belonging to the permissioned User Group; optional

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
view_report	boolean	true|false	Allow to View Report - permission to view the reports of messages sent by Users belonging to the permissioned User Group; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/UserGroupUgp/Set

Request:

```
<SetUserGroupUgp xmlns="http://www.hiplink.com/">
  <SetUserGroupUgpRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <other_user_group>string</other_user_group>
    <user_group>string</user_group>
    <modify_user>bool</modify_user>
    <view_report>bool</view_report>
  </SetUserGroupUgpRequest>
</SetUserGroupUgp>
```

Response:

```
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <SetUserGroupUgpResponse xmlns="http://www.hiplink.com/">
      <SetUserGroupUgpResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
      </SetUserGroupUgpResult>
    </SetUserGroupUgpResponse>
```

Example

```
<SetUserGroupUgp xmlns="http://www.hiplink.com/">
      <SetUserGroupUgpRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <other_user_group>usrsend</other_user_group>
        <user_group>sysoper</user_group>
        <modify_user>true</modify_user>
        <view_report>false</view_report>
      </SetUserGroupUgpRequest>
    </SetUserGroupUgp>
```

©HipLink Software 2013. All Rights Reserved.

ADD RECEIVER

Parameters

Name	Type	Value	Comment
receiver_name	string	receiver name, e.g., Test Rec	Receiver Name; mandatory
department_name	string	department name, e.g., Test Department	Department; mandatory (only if 'Departments' are enabled by the License Key)
description	string	receiver description, e.g., Test Receiver	Description optional
pin	string	receiver PIN, e.g., 1231323	PIN; mandatory
carrier	string	carrier name, e.g., Test	Carrier; mandatory
alternate_pin	string	alternate PIN, e.g., 134539	Backup PIN; optional
alternate_carrier	string	alternate carrier, e.g., Other Carrier	Backup Carrier; optional
last_name	string	Last Name, e.g., Chad	Last Name; optional
first_name	string	First Name, e.g., Adams	Fist name; optional
email_address	string	Email Address, e.g., Adams@test.com	Email Address; optional
email_failover	boolean	Email failover true|false	Email Failover; optional
email_cc	boolean	Email CC true|false	Email CC; optional
timezone	int	User Time Zone	Time Zone; mandatory
is_alpha	boolean	Alpha charcters true|false	Keep alpha chars; optional
device_type	int	Device Type	Device Type; mandatory
security_code	string	Security Code	Security Code; optional
account_name	string	Account name e.g., admin	Account Name; optional
account_password	string	Account Password e.g., admin	Account Password; optional
allow_edit	boolean	Allow edit true|false	Edit Receiver; optional
allow_schedule	boolean	Allow schedule true|false	update receiver schedule info; optional
cover_by_receiver	string	Cover by receiver	Cover By Receiver; optional
disabled	boolean	Disabled true|false	Receiver Status; optional
enable_voice	boolean	Enable voice	Voice enabled; optional

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
		true|false	
voice_number	string	Voice number, e.g., 1234321	Voice Phone Number; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Receiver/Add

Request:

```
<AddReceiver xmlns="http://www.hiplink.com/">
  <AddReceiverRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <account_name>string</account_name>
    <account_password>string</account_password>
    <allow_edit>bool</allow_edit>
    <allow_schedule>bool</allow_schedule>
    <alternate_carrier />
    <alternate_pin />
    <carrier>string</carrier>
    <cover_by_receiver />
    <department_name>string</department_name>
    <description>string</description>
    <device_type>string</device_type>
    <disabled>bool</disabled>
    <email_address>string</email_address>
    <enable_cc>bool</enable_cc>
    <enable_failover>bool</enable_failover>
    <enable_voice>bool</enable_voice>
    <first_name>bool</first_name>
    <is_alpha>bool</is_alpha>
    <last_name />
    <pin>string</pin>
    <receiver_name>string</receiver_name>
    <security_code>string</security_code>
    <timezone>string</timezone>
    <voice_number/>
    <assigned_owner>string</assigned_owner>
    <enable_adv_messaging>bool</enable_adv_messaging>
  </AddReceiverRequest>
</AddReceiver>
```

Response:

```
<AddReceiverResponse xmlns="http://www.hiplink.com/">
  <AddReceiverResult>
    <response_message>string</response_message>
```

©HipLink Software 2013. All Rights Reserved.

```
        <response_code>string</response_code>
      </AddReceiverResult>
    </AddReceiverResponse>
```

Example

```
<AddReceiver xmlns="http://www.hiplink.com/">
      <AddReceiverRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <account_name>test</account_name>
        <account_password>test</account_password>
        <allow_edit>true</allow_edit>
        <allow_schedule>true</allow_schedule>
        <alternate_carrier />
        <alternate_pin />
        <carrier>fax</carrier>
        <cover_by_receiver />
        <department_name>abc</department_name>
        <description>test</description>
        <device_type>fax</device_type>
        <disabled>false</disabled>
        <email_address>t@t.com</email_address>
        <enable_cc>false</enable_cc>
        <enable_failover>false</enable_failover>
        <enable_voice>false</enable_voice>
        <first_name>test</first_name>
        <is_alpha>true</is_alpha>
        <last_name />
        <pin>test</pin>
        <receiver_name>umair</receiver_name>
        <security_code>23456</security_code>
        <timezone>Server Time</timezone>
        <voice_number />
        <assigned_owner>admin</assigned_owner>
  <enable_adv_messaging>true</enable_adv_messaging>
      </AddReceiverRequest>
    </AddReceiver>
```

DELETE RECEIVER

Parameters

Name	Type	Value	Comment
receiver_name	string	receiver name, e.g., Test Receiver	Receiver Name; mandatory

REST Sample

©HipLink Software 2013. All Rights Reserved.

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Receiver/Delete

Request:

```
<DeleteReceiver xmlns="http://www.hiplink.com/">
  <DeleteReceiverRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <receiver_name>string</receiver_name>
  </DeleteReceiverRequest>
</DeleteReceiver>
```

Response:

```
<DeleteReceiverResponse xmlns="http://www.hiplink.com/">
  <DeleteReceiverResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </DeleteReceiverResult>
</DeleteReceiverResponse>
```

Example

```
<DeleteReceiver xmlns="http://www.hiplink.com/">
      <DeleteReceiverRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <receiver_name>umair</receiver_name>
      </DeleteReceiverRequest>
    </DeleteReceiver>
```

QUERY RECEIVER

Parameters

Name	Type	Value	Comment
name_filter	string	receiver name, e.g. Test Receiver	Receiver Name; mandatory (the name can contain “*”)
dept_filter	string	Department, e.g,sales Department	Department Name; mandatory (the name can contain “*”)

REST Sample

©HipLink Software 2013. All Rights Reserved.

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Receiver/Query

Request:

```
<QueryReceiver xmlns="http://www.hiplink.com/">
  <QueryReceiverRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <name_filter>string</name_filter>
    <dept_filter>string</dept_filter>
  </QueryReceiverRequest>
</QueryReceiver>
```

Response:

```
<QueryReceiverResponse xmlns="http://www.hiplink.com/">
  <QueryReceiverResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <receiver>
      <string></string>
      <string></string>
    </receiver>
  </QueryReceiverResult>
</QueryReceiverResponse>
```

Example 1

```
<QueryReceiver xmlns="http://www.hiplink.com/">
      <QueryReceiverRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <name_filter>*</name_filter>
        <dept_filter>abc</dept_filter>
      </QueryReceiverRequest>
    </QueryReceiver>
```

ADD GROUP

Parameters

Name	Type	Value	Comment
group_name	string	group name, e.g., Test Grp	Group Name; mandatory
group_type	string	Department, e.g., Sales department	Group Type; mandatory
department_name	string	receiver | escalation | onduty | followme | rotate	Department Name; mandatory (only if 'Departments' are

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
			enabled by the License Key)
topic	string	Topic	optional
description	string	group description, e.g., Test Description	Description; optional
rotating	boolean	true|false	
owner_settings_enable	boolean	Owner Settings true|false	Enable owner settings; optional
owner_do_alert	boolean	Alert Owner true|false	Alert Owner of Changes; optional
owner_limit_access	boolean	Owner limit true|false	Owner limit; optional
owner_user_name	string	Owner User Name, e.g., SysAdmin	Owner Name; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Group/Add

Request:

```
<AddGroup xmlns="http://www.hiplink.com/">
  <AddGroupRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <department_name>string</department_name>
    <description>string</description>
    <group_name>string</group_name>
    <group_type>string</group_type>
    <owner_user_name>string</owner_user_name>
    <topic />
    <owner_do_alert>bool</owner_do_alert>
    <owner_limit_access>bool</owner_limit_access>
    <owner_settings_enable>bool</owner_settings_enable>
    <rotating>bool</rotating>
  </AddGroupRequest>
</AddGroup>
```

Response:

```
<AddGroupResponse xmlns="http://www.hiplink.com/">
  <AddGroupResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </AddGroupResult>
```

©HipLink Software 2013. All Rights Reserved.

```
    </AddGroupResponse>
```

Example

```
<AddGroup xmlns="http://www.hiplink.com/">
      <AddGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <department_name>abc</department_name>
        <description>test</description>
        <group_name>new grp</group_name>
        <group_type>broadcast</group_type>
        <owner_user_name>admin</owner_user_name>
        <topic />
        <owner_do_alert>true</owner_do_alert>
        <owner_limit_access>false</owner_limit_access>
        <owner_settings_enable>true</owner_settings_enable>
        <rotating>false</rotating>
      </AddGroupRequest>
    </AddGroup>
```

DELETE GROUP

Parameters

Name	Type	Value	Comment
group_name	string	group name, e.g., Test Group	Group Name; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Group/Delete

Request:

```
    <DeleteGroup xmlns="http://www.hiplink.com/">
      <DeleteGroupRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <group_name>string</group_name>
      </DeleteGroupRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
</DeleteGroup>
```

Response:

```
<DeleteGroupResponse xmlns="http://www.hiplink.com/">
  <DeleteGroupResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </DeleteGroupResult>
</DeleteGroupResponse>
```

Example

POST /HipLinkService.asmx HTTP/1.1
Host: localhost
Content-Type: text/xml; charset=utf-8
Content-Length: length
SOAPAction: "DeleteGroup"

```
<?xml version="1.0" encoding="utf-8"?>
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
  <soap:Body>
    <DeleteGroup xmlns="http://www.hiplink.com/">
      <DeleteGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>new grp</group_name>
      </DeleteGroupRequest>
    </DeleteGroup>
  </soap:Body>
</soap:Envelope>
```

QUERY GROUP

Parameters

Name	Type	Value	Comment
name_filter	string	group name, e.g., Test Grp	Group Name (the name can contain “*”); mandatory
dept_filter	string	Department name, e.g., Sales department	Department Name (the name can contain “*”); mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Group/Query

©HipLink Software 2013. All Rights Reserved.

Request:

```
<QueryGroup xmlns="http://www.hiplink.com/">
  <QueryGroupRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <name_filter>string</name_filter>
    <dept_filter>string</dept_filter>
  </QueryGroupRequest>
</QueryGroup>
```

Response:

```
<QueryGroupResponse xmlns="http://www.hiplink.com/">
  <QueryGroupResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <group>
      <string></string>
      <string></string>
    </group>
  </QueryGroupResult>
</QueryGroupResponse>
```

Example 1

```
<QueryGroup xmlns="http://www.hiplink.com/">
      <QueryGroupRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <name_filter>*</name_filter>
        <dept_filter>abc</dept_filter>
      </QueryGroupRequest>
    </QueryGroup>
```

ADD A GROUP MEMBER

Parameters

Name	Type	Value	Comment
group_name	string	group name, e.g., IT Receiver Group	Group Name; mandatory
member_name	String	Member name	Member name; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

©HipLink Software 2013. All Rights Reserved.

URL: /axis2/services/hiplink/GroupMember/Add

Request:

```
<AddGroupMember xmlns="http://www.hiplink.com/">
  <AddGroupMemberRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <group_name>string</group_name>
    <member_name>string</member_name>
  </AddGroupMemberRequest>
</AddGroupMember>
```

Response:

```
<AddGroupMemberResponse xmlns="http://www.hiplink.com/">
  <AddGroupMemberResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </AddGroupMemberResult>
</AddGroupMemberResponse>
```

Example

```
<AddGroupMember xmlns="http://www.hiplink.com/">
      <AddGroupMemberRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>edqs</group_name>
        <member_name>abbott, mark</member_name>
      </AddGroupMemberRequest>
</AddGroupMember>
```

DELETE GROUP MEMBER

Parameters

Name	Type	Value	Comment
group_name	string	group name, e.g. IT Receiver Group	Group Name; mandatory
member_name	String	Member name	Member name; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

©HipLink Software 2013. All Rights Reserved.

URL: /axis2/services/hiplink/GroupMember/Delete

Request:

```
<DeleteGroupMember xmlns="http://www.hiplink.com/">
  <DeleteGroupMemberRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <group_name>string</group_name>
    <member_name>string</member_name>
  </DeleteGroupMemberRequest>
</DeleteGroupMember>
```

Response:

```
<DeleteGroupMemberResponse xmlns="http://www.hiplink.com/">
  <DeleteGroupMemberResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </DeleteGroupMemberResult>
</DeleteGroupMemberResponse>
```

Example

```
<DeleteGroupMember xmlns="http://www.hiplink.com/">
      <DeleteGroupMemberRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <group_name>edqs</group_name>
        <member_name>abbott, mark</member_name>
      </DeleteGroupMemberRequest>
</DeleteGroupMember>
```

SEND STANDARD MESSAGE

Parameters

Name	Type	Value	Comment
recipient_name	string	recipient name, e.g., Bill Jones	Recipient Name; mandatory
message	String	message body e.g., hello world!	Message - the message body; mandatory
include_timestamp	boolean	add name and time stamp true| false	Add Name/Time Stamp - defines if the sender name and timestamp should be automatically appended to the message; optional
include_sender	boolean	Include sender	Include sender; optional
hi_priority	boolean	priority	Message priority; optional

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
include_job_id	boolean	Include Job Id	Include Job ID; optional
email_receipt	boolean	Email receipt	Email receipt; optional
job_id	String	custom Job ID, e.g., ID345	Custom Job ID - the Job ID of the message. This Job ID overwrites the one generated by the HipLink server and which is guaranteed to be unique. If this feature is used, then the User has the responsibility for the Job ID uniqueness; optional
sender	String	sender name, e.g., admin	Sender - specifies the name of the sender of the message. This name is used to replace the CLI (i.e., Non GUI) User in the reports. It can be set to any string because the CLI permissions are based on the Non GUI User settings (i.e., IP address of the machine where the CLI is installed); optional
subject	String	Subject	Message Subject; optional
delivery_report	String	Delivery report	Message Delivery Report; optional

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/StdMessage/Send

Request:

```
<SendStdMessage xmlns="http://www.hiplink.com/">
  <StandardSendRequest xmlns="">
    <credential>
      <username>admin</username>
      <password>admin</password>
    </credential>
    <recipient_name>string</recipient_name>
    <message>string</message>
    <include_timestamp>boolean</include_timestamp>
    <include_sender>boolean</include_sender>
    <hi_priority>boolean</hi_priority>
    <include_job_id>boolean</include_job_id>
    <email_receipt>boolean</email_receipt>
    <job_id>string</job_id>
    <sender>string</sender>
    <subject>string</subject>
    <delivery_report>string</delivery_report>
  </StandardSendRequest>
</SendStdMessage>
```

Response:

```
<SendStdMessageResponse xmlns="http://www.hiplink.com/">
```

©HipLink Software 2013. All Rights Reserved.

```
        <SendStdMessageResult>
          <response_message>string</response_message>
          <response_code>string</response_code>
          </SendStdMessageResult>
    </SendStdMessageResponse>
```

Example

```
<SendStdMessage xmlns="http://www.hiplink.com/">
      <StandardSendRequest xmlns="">
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <recipient_name>Abbey</recipient_name>
        <message>test message</message>
        <include_timestamp>true</include_timestamp>
        <include_sender>false</include_sender>
        <hi_priority>true</hi_priority>
        <include_job_id>true</include_job_id>
        <email_receipt>true</email_receipt>
        <job_id>1509_1</job_id>
        <sender>test</sender>
        <subject>test message subject</subject>
        <delivery_report>string</delivery_report>
      </StandardSendRequest>
    </SendStdMessage>
```

SEND QUICK MESSAGE

Parameters

Name	Type	Value	Comment
include_timestamp	boolean	add name and time stamp true|false	Add Name/Time Stamp - defines if the sender name and timestamp should be automati-cally appended to the message; optional
include_sender	boolean	Include sender true|false	Include sender; optional
hi_priority	boolean	High priority true|false	Message Priority; optional
include_job_id	boolean	Include Job Id true|false	optional
subject	string	Message subject e.g., Test sub	Message Subject; optional
message	string	message body e.g., hello world!	Message - the message body; mandatory
authorization_code	string	Authorization code e.g., 3456	Message Authorization Code; optional
carrier_name	string	carrier name e.g., Verizon	Carrier - the Carrier used by the Receiver; mandatory

©HipLink Software 2013. All Rights Reserved.

Name	Type	Value	Comment
pin	string	recipient PIN e.g., 1234	PIN - the pin number of the Receiver to which the mes-sage will be sent; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/QuickMessage/Send

Request:

```
<SendQuickMessage xmlns="http://www.hiplink.com/">
  <SendQuickMessageRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <carrier_name>string</carrier_name>
    <hi_priority>bool</hi_priority>
    <include_job_id>bool</include_job_id>
    <include_sender>bool</include_sender>
    <include_timestamp>bool</include_timestamp>
    <message>string</message>
    <pin>string</pin>
    <authorization_code>string</authorization_code>
    <subject>string</subject>
  </SendQuickMessageRequest>
</SendQuickMessage>
```

Response:

```
<SendQuickMessageResponse xmlns="http://www.hiplink.com/">
  <SendQuickMessageResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </SendQuickMessageResult>
</SendQuickMessageResponse>
```

Example

```
<SendQuickMessage xmlns="http://www.hiplink.com/">
      <SendQuickMessageRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <carrier_name>hnp two-way carrier 1</carrier_name>
        <hi_priority>true</hi_priority>
        <include_job_id>true</include_job_id>
        <include_sender>true</include_sender>
        <include_timestamp>true</include_timestamp>
        <message>test message for hnp</message>
```

©HipLink Software 2013. All Rights Reserved.

```
      <pin>hnp</pin>
      <authorization_code>1234</authorization_code>
      <subject>test subject</subject>
    </SendQuickMessageRequest>
  </SendQuickMessage>
```

QUERY MESSAGE STATUS

Parameters

Name	Type	Value	Comment
job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/MessageStatus/Query

Request:

```
  <QueryMessageStatus xmlns="http://www.hiplink.com/">
    <QueryMessageStatusRequest>
      <credential>
        <username>string</username>
        <password>string</password>
      </credential>
      <job_id>string</job_id>
    </QueryMessageStatusRequest>
  </QueryMessageStatus>
```

Response:

```
  <QueryMessageStatusResponse xmlns="http://www.hiplink.com/">
    <QueryMessageStatusResult>
      <response_message>string</response_message>
      <response_code>string</response_code>
      <status>string</status>
      <failed_reason>string</failed_reason>
    </QueryMessageStatusResult>
  </QueryMessageStatusResponse>
```

Example

```
<QueryMessageStatus xmlns="http://www.hiplink.com/">
      <QueryMessageStatusRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <job_id>2031_1</job_id>
```

©HipLink Software 2013. All Rights Reserved.

```
    </QueryMessageStatusRequest>
  </QueryMessageStatus>
```

CONFIRM MESSAGE

Parameters

Name	Type	Value	Comment
job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory
comments	string	Comments, e.g., Test Comment	Any comments about the message; optional
custom_status	String	Message Status, e.g., Confirm or Reject	Status of the Message as confirm or reject; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/Message/Confirm

Request:

```
<ConfirmMessage xmlns="http://www.hiplink.com/">
  <ConfirmMessageRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <job_id>string</job_id>
    <comments>string</comments>
    <custom_status>string</custom_status>
  </ConfirmMessageRequest>
</ConfirmMessage>
```

Response:

```
<ConfirmMessageResponse xmlns="http://www.hiplink.com/">
  <ConfirmMessageResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </ConfirmMessageResult>
</ConfirmMessageResponse>
```

Example

```
<ConfirmMessage xmlns="http://www.hiplink.com/">
      <ConfirmMessageRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
```

©HipLink Software 2013. All Rights Reserved.

```
    </credential>
    <job_id>2009_1</job_id>
    <comments>test</comments>
    <custom_status>confirmed</custom_status>
  </ConfirmMessageRequest>
</ConfirmMessage>
```

QUERY MESSAGE HISTORY

Parameters

Name	Type	Value	Comment
job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/MessageHistory/Query

Request:

```
<QueryMessageHistory xmlns="http://www.hiplink.com/">
  <QueryMessageHistoryRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <job_id>string</job_id>
  </QueryMessageHistoryRequest>
</QueryMessageHistory>
```

Response:

```
<QueryMessageHistoryResponse xmlns="http://www.hiplink.com/">
  <QueryMessageHistoryResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
    <message>
      string
    </message>
    <failed_reason>string</failed_reason>
    <receiver_name>string</receiver_name>
    <department_name>string</department_name>
    <job_history>
      <history_item>
        <timestamp>string</timestamp>
```

©HipLink Software 2013. All Rights Reserved.

```
            <pin>string</pin>
            <carrier>string</carrier>
            <status>string</status>
          </history_item>
        </job_history>
      </QueryMessageHistoryResult>
    </QueryMessageHistoryResponse>
```

Example

```
<QueryMessageHistory xmlns="http://www.hiplink.com/">
      <QueryMessageHistoryRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <job_id>2009_1</job_id>
      </QueryMessageHistoryRequest>
    </QueryMessageHistory>
```

QUERY GROUP MESSAGE STATUS

Parameters

Name	Type	Value	Comment
major_job_id	string	Job ID, e.g., 123_1	Custom Job ID - the Job ID of the message for which the status is queried; mandatory

REST Sample

The following is a sample REST request and response. The placeholders shown need to be replaced with actual values.

URL: /axis2/services/hiplink/GroupMessageStatus/Query

Request:

```
    <QueryGroupMessageStatus xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusRequest>
        <credential>
          <username>string</username>
          <password>string</password>
        </credential>
        <major_job_id>string</major_job_id>
      </QueryGroupMessageStatusRequest>
    </QueryGroupMessageStatus>
```

Response:

```
    <QueryGroupMessageStatusResponse xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusResult>
        <response_message>string</response_message>
        <response_code>string</response_code>
```

©HipLink Software 2013. All Rights Reserved.

```
        <message_statuses>
          </message_status>
          <message_status>
            <timestamp>string</timestamp>
            <job_id>string</job_id>
            <receiver_name>string</receiver_name>
            <status>string</status>
            <failed_reason>string</failed_reason>
          </message_status>
        </message_statuses>
      </QueryGroupMessageStatusResult>
    </QueryGroupMessageStatusResponse>
```

Example

```
<QueryGroupMessageStatus xmlns="http://www.hiplink.com/">
      <QueryGroupMessageStatusRequest>
        <credential>
          <username>admin</username>
          <password>admin</password>
        </credential>
        <major_job_id>2032_1</major_job_id>
      </QueryGroupMessageStatusRequest>
    </QueryGroupMessageStatus>
```

SET USER GROUP RESPONSE ACTION PERMISSION

Parameters

Name	Type	Value	Comment
user_group	string	Action name , e.g., test	User Group name for which permissions are applied; mandatory
Response_Action_Name	string	Parameter name, e.g., test	Reponse action name; mandatory
execute_action	boolean	Parameter name, e.g., true	Enable or disable the response action; mandatory

REST Sample

URL: /axis2/services/hiplink/UserGroup/ResponseActionPerm/Set

Request:

```
<SetUserGroupResponseActionPerm xmlns="http://www.hiplink.com/">
  <SetUserGroupResponseActionPermRequest>
    <credential>
      <username>string</username>
      <password>string</password>
    </credential>
    <user_group>string</user_group>
    <response_action_name>string</response_action_name>
    <execute_action>string</execute_action>
  </SetUserGroupResponseActionPermRequest>
```

©HipLink Software 2013. All Rights Reserved.

```
</SetUserGroupResponseActionPerm>
```

Response:

```
<SetUserGroupResponseActionPermResponse xmlns="http://www.hiplink.com/">
  <SetUserGroupResponseActionPermResult>
    <response_message>string</response_message>
    <response_code>string</response_code>
  </SetUserGroupResponseActionPermResult>
</SetUserGroupResponseActionPermResponse>
```

Example

```
<SetUserGroupResponseActionPerm xmlns="http://www.hiplink.com/">
  <SetUserGroupResponseActionPermRequest>
    <credential>
      <username>admin</username>
      <password>admin</password>
    </credential>
    <user_group>taimoor all</user_group>
    <response_action_name>GetUserData</response_action_name>
    <execute_action>true</execute_action>
  </SetUserGroupResponseActionPermRequest>
</SetUserGroupResponseActionPerm>
```

©HipLink Software 2013. All Rights Reserved.

Web Sign-up Javascript API

Introduction

The HipLink Web Sign-up JavaScript API is session less programming interface for the JavaScript language. This Interface provides functions to make transactions to the active Web Sign-up in HipLink. This document is a complete function reference of the Web Sign-up JavaScript API.

The source file HLWebSignupAPI.js can be found in:

- On IIS installation: hiplink_install_directory/server/htdocs/includes/
- On Apache Installation: hiplink_install_directory/apache/htdocs/includes/

Who Should Read This?

This document requires prior knowledge of JavaScript language. More over there is extensive use of JSON Objects in this API. JSON is a JavaScript Object Notation which specifies how JSON Objects can be created. You can view more information on JSON at the following web site:

- http://www.json.org/

Pre-Requisite

The only pre-requisite for using Web Sign-up API is to include PrototypeJS (prototype.js) which is included in the HipLink installation. PrototypeJS is a JavaScript framework which is being used for AJAX calls to the server.

To include both, the PrototyepJS framework file and Web Signup JS API file, add the following tags in the 'head' tag:

- <script LANGUAGE="JavaScript" SRC="/includes/prototype.js"></script>
- <script LANGUAGE="JavaScript" SRC="/includes/HLWebSignupAPI.js"></script>

Variables

- HLWebSignup.Initialized type: Boolean
- HLWebSignup.Enabled type: Boolean
- HLWebSignup.Config type: JSON Object

Functions

All the functions that return a JSON Object have a member named status (e.g. obj.status) the value of which is either OK or ERROR. If the status is ERROR, there will be a member named 'message' (e.g. obj.message) that will contain an explanation as to why the error occurred.

Common JSON Object members:

Member	Type	Optional	Value

©HipLink Software 2013. All Rights Reserved.

Obj.status	String	No	OK or ERROR
Obj.message	String	Yes	In case of ERROR status, this field becomes mandatory.

All function parameters are mandatory in every function. Functions which take a JSON object as an argument may have optional members in it.

- WeSU_Initialize()
- Arguments: None
- Returns: JSON Object

Explanation:

This function initializes the Web Sign-up configuration object by calling the WeSU_Config function. It checks whether the load was successful or not. If the configuration object is fetched successfully, HLWebSignup.Enabled is set to true and a copy of the configuration object is saved in HLWebSignup.Config. When the function completes execution HLWebSignup.Initialized is set to true irrespective of the call being successful or unsuccessful.

- WeSU_Config()
- Arguments: None
- Returns: JSON Object

Explanation:

This function fetches the Web Sign-up configuration, copies it to HLWebSignup.Config, and returns the configuration object.

If the configuration is fetched successfully the returned JSON Object (here referenced as Obj) will have the following members:

Member	Type	Optional	Value
Obj.config	JSON Object	No	The Web Sign-up configuration object that contains various members
Obj.config.name	String	No	Name of the Web Sign-up
Obj.config.description	String	No	Description of the Web Sign-up
Obj.config.enable	JSON Object	No	Object containing different device types allowed
Obj.config.enable.voice	Boolean	No	Voice devices allowed
Obj.config.enable.text	Boolean	No	Text devices allowed
Obj.config.maxDevices	Number	No	Number of devices allowed
Obj.config.carriers	Array String	No	An array of allowed carriers to be used with Text devices
Obj.config.topics	Array JSON Object	No	An array of topic objects
Obj.config.topics[i].name	String	No	Name of the topic
Obj.config.topics[i].description	String	No	Description of the topic explaining what kind of alerts will be received on this

©HipLink Software 2013. All Rights Reserved.

Member	Type	Optional	Value
Obj.config.characterisics	String	No	An array of characteristics
Obj.config.address	Array String	Yes	An array of address field IDs
Obj.config.addressDisplay	Array String	Yes	An array of address field displayable labels to be used with Obj.config.address array.

- WeSU_Register(user, verifytext)
- Arguments: user JSON object
- Verifytext string
- Returns: JSON Object

Explanation:

This function takes the user object and the verifytext (verification text) as an argument and returns registration AJAX call response.

The user object provided to the function must have the following format:

Member	Type	Optional	Value
user.name	JSON Object	No	
user.name.first	String	No	First name of the user
user.name.last	String	No	Last name of the user
user.email	String	No	Email of the user that will be used as a username for the Web Sign-up
user.password	String	No	User specified password
user.topics	Array String	Yes	An array of selected topic names
user.characteristics	Array String	Yes	An array of selected characteristic names
user.device	Array JSON Object	No	An array of device objects, must contain at least one device
user.device[i].type	String	No	'text' or 'voice'
user.device[i].pin	String	No	User specified pin or phone number
user.device[i].carrier	String	No	The carrier name selected for the pin. For 'voice' type devices, no carrier needs to be specified. If any carrier is specified, it will be ignored
user.address	JSON Object	Yes	An Object of address IDs and their values
user.address.<Address field ID>	String	No	The '<>' specify that the member name is determined buy the ID fetched in the config object and the value is the user specified value for it

- WeSU_Confirm(username, passkey)
- Arguments: username string
  - Passkey string
- Returns: JSON Object

Explanation:

©HipLink Software 2013. All Rights Reserved.

This function takes user name (email) and pass key (a string of alphabets and digits) and returns the confirmation AJAX call response.

- WeSU_Login(username, password)
- Arguments: username string
  - Password string
- Returns: JSON Object

Explanation:

This function takes user name (email) and password and returns the login AJAX call response. The Database is updated by the requested posted for the last login time.

- WeSU_UserInfo(username, password)
- Arguments: username string
  - Password string
- Returns: JSON Object

Explanation:

This function takes user name (email) and password and returns the userinfo AJAX call response. The response is a JSON Object with various members that form part of the user info.

If the user info is fetched successfully the returned JSON Object (here referenced as Obj) will have the following members:

Member	Type	Optional	Value
Obj.user	JSON Object	No	The user info object
Obj.user.email	String	No	The user email that will be the username and used for correspondence
Obj.user.name	JSON Object	No	The user first and last name object
Obj.user.name.first	String	No	User's first name
Obj.user.name.last	String	No	User's last name
Obj.user.charac	Array String	Yes	The user's selected characteristics
Obj.user.topic	Array String	Yes	The user's selected topics
Obj.user.device	Array JSON Object	No	An array of saved user devices
Obj.user.device[i].pin	String	No	PIN of the device
Obj.user.device[i].type	String	No	'text' or 'voice'
Obj.user.device[i].carrier	String	Yes	In case of a 'text' device, this member is mandatory
Obj.user.lastlogin	String	No	The last login time of the user
Obj.user.joined	String	No	The date/time the user joined
Obj.user.address	Array JSON Object	Yes	The saved address array of the user

©HipLink Software 2013. All Rights Reserved.

Member	Type	Optional	Value
Obj.user.address[i].name	String	No	The Address field ID of the address part
Obj.user.address[i].value	String	No	The saved address value

- WeSU_Update(user, verifytext)
- Arguments: user JSON object
  - Verifytext string
- Returns: JSON Object

Explanation:

This function is similar to the WeSU_Register function call and takes the same user object and the verifytext (verification text) as an argument and returns the user info update AJAX call response.

The user object provided to the function must have the following format:

Member	Type	Optional	Value
user.name	JSON Object	No	
user.name.first	String	No	First name of the user
user.name.last	String	No	Last name of the user
user.email	String	No	Email of the user that will be used as a username for the Web Sign-up
user.password	String	No	User specified password
user.topics	Array String	Yes	An array of selected topic names
user.characteristics	Array String	Yes	An array of selected characteristic names
user.device	Array JSON Object	No	An array of device objects. Must contain at least one device
user.device[i].type	String	No	'text' or 'voice'
user.device[i].pin	String	No	User specified pin or phone number
user.device[i].carrier	String	No	The carrier name selected for the pin. For 'voice' type devices, no carrier needs to be specified. If any carrier is specified, it will be ignored
user.address	JSON Object	Yes	An Object containing address IDs as members and the user specified values
user.address.<Address field ID>	String	No	The '<>' denotes that the member name is determined by the ID fetched in the config object by the 'WeSU_Initialize' or 'WeSU_Config' functions and the value is the user specified value for it
User.newpassword	String	Yes	New password to be set for the user

©HipLink Software 2013. All Rights Reserved.

- WeSU_ForgotPassword(username)
- Arguments: username string
- Returns: JSON Object

Explanation:

This function takes user name (email) and returns the “forgot password” AJAX call response. The user record with the corresponding email is updated for the new password and the password is mailed to the user to that email address.

- WeSU_Cancel(username, password, verifytext)
- Arguments: username string
  - Password string
- Returns: JSON Object

Explanation:

This function takes user name (email), password the verifytext (verification text) and returns the cancel AJAX call response. The cancel AJAX request cancels the user subscription and the user record is deleted from the database.

- WeSU_AuthImage(imgID)
- Arguments: username string
  - Password string
- Returns: JSON Object

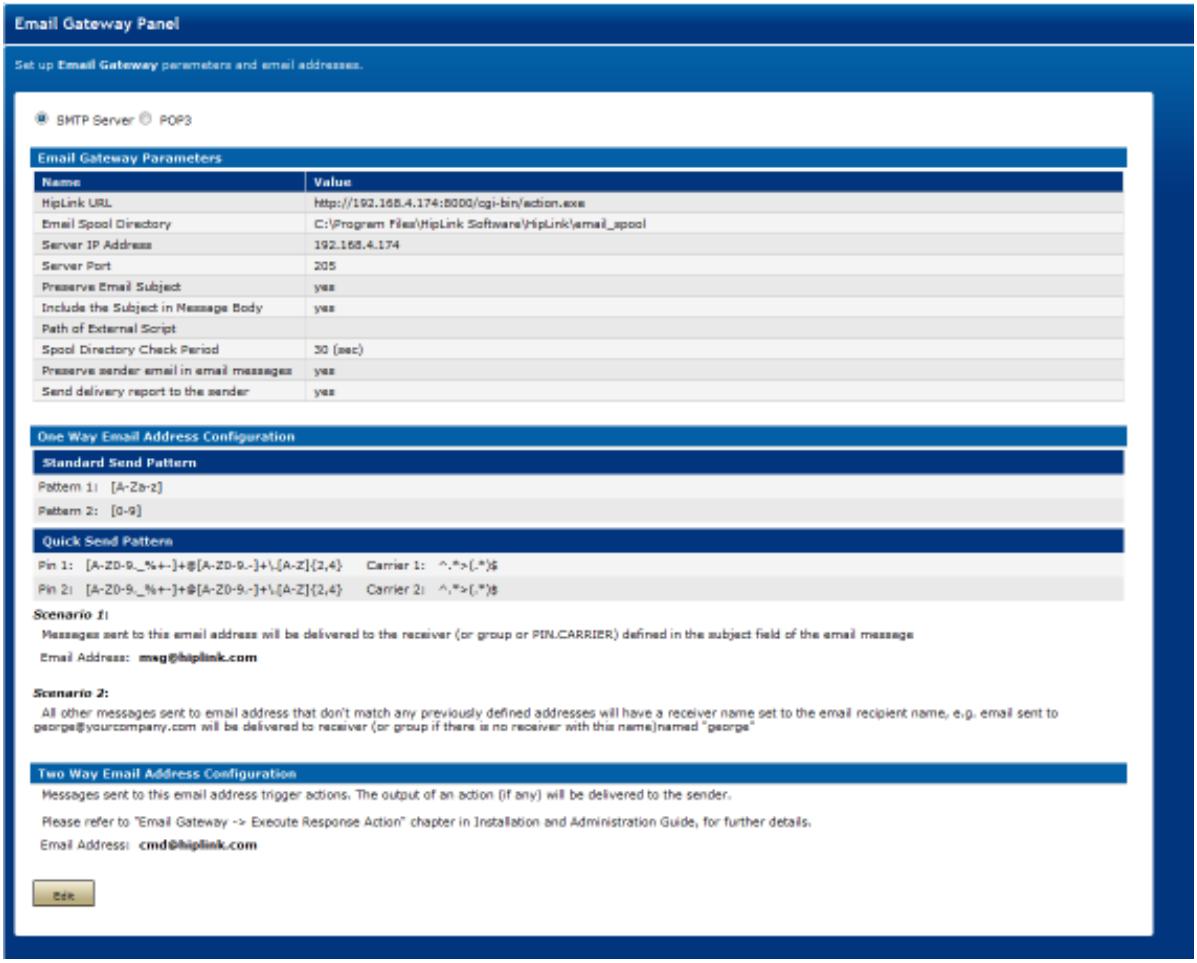
Explanation:

This function takes the id of an img HTML tag and loads a new verification image from the server by making a call to /cgi-bin/soter.cgi. This call to soter.cgi causes in the creation or updation of SOTER-ALPHA and SOTER-BRAVO cookies. These Cookies are necessary for WeSU_Register, WeSU_Update and WeSU_Cancel function calls. The text seen on this image must be provided in the ‘verifytext’ argument of these functions.

©HipLink Software 2013. All Rights Reserved.

Email Gateway

The HipLink Email Gateway is enabled through your License Key. Please check the License Key Panel from the Settings Menu to see that this feature is enabled for your version of HipLink. If it is not and you want to use the Email Gateway you need a new License Key. Please contact HipLink Tech Support at support@hiplink.com for assistance. Set up the Email Gateway. The Email Gateway supports both the SMTP and POP3 protocols. In order to work properly, the Email Gateway has to be set up properly. Please refer to the Hip-Link Installation and Administration Guide for details about setting up the Email Gateway.



Email Gateway Panel

SMTP Server Email Gateway Parameters

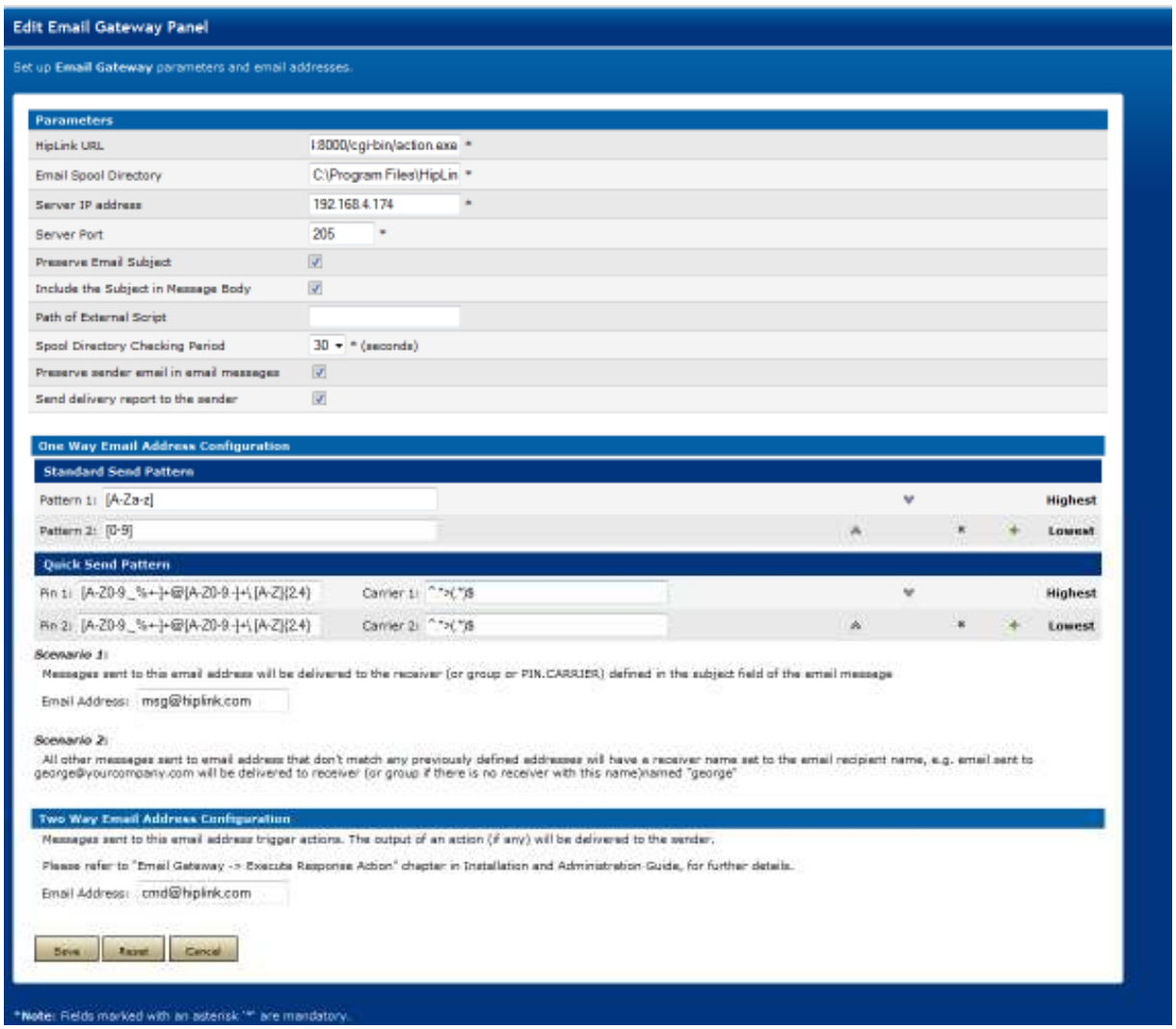
To set the SMTP Server Email Gateway parameters:

1. From the Settings menu, click Email Gateway on the left navigation bar.

2. On the Email Gateway Panel, click the SMTP Server radio button (default).

3. Click the Edit button to reach the Edit Email Gateway Panel.

©HipLink Software 2013. All Rights Reserved.



Edit Email Gateway Panel - SMTP Server

The SMTP Server Email Gateway parameters that can be set are:

- **HipLink URL** - specifies the URL of the HipLink instance used by the SMTP Server Email Gateway to send messages.
- **Email Spool Directory** - specifies the directory where the emails received by the gateway are stored.
- **Server IP Address**- specifies the IP address of the machine that hosts the SMTP Server Email Gateway.
- **Server Port** - specifies the port of the SMTP Server Email Gateway.
- **Preserve Email Subject** - if checked, the email dispatched to the recipient will have the subject as defined in the email; if un-checked, the Default SMTP Subject defined in Global Settings would be used.
- **Include the Subject in Message Body** - if checked, the subject of the email would be included in the message body.
- **Path of External Script** – specifies the path of the external script to be executed when a message is sent by Email Gateway.

©HipLink Software 2013. All Rights Reserved.

- **Spool Directory Checking Period** - specifies the time period at which the SMTP Server Email Gateway is checking the spool for new messages.
- **Preserve sender email in email messages** – the email address received by the recipient would contain email address of the original sender.
- **Send delivery report to the sender** – if checked, the delivery report would be sent back to the sender.

Enter the one-way messaging email address in the One-Way Email Address Configuration section.

Enter the two-way messaging email address in the Two-Way Email Address Configuration section.

Click the Save button to submit your changes and return to the Email Gateway Panel, Reset button to fill in the previous values, or Cancel button to return without saving.

Note: Any changes to the Email Gateway settings require the Email Gateway Service to be restarted.

- One-Way Email Address Configuration:
- Standard Send Pattern:
- Define a pattern(s) for the receiver name to send a standard message.
- Quick Send Pattern:
- Define a pattern(s) for the receiver pin and carrier to send a quick message.

The one-way SMTP Server Email Gateway is designed to work following two different scenarios of processing the emails.

Scenario 1: Messages sent to the generic email address defined for the first scenario are delivered to the recipient (Receiver or Group) which is defined in the Subject field of the email message. The Receiver or Group should be already defined in the HipLink database. For example, if there is an email sent to HipLink@company.com with the subject Joe Smith, then HipLink will send the body of the message to the receiver Joe Smith.

Scenario 2: Messages sent to email addresses other than those previously defined for Scenario 1 are sent to the recipient (Receiver or Group) that matches the email recipient name. The Receiver or Group should be already defined in the HipLink database. For example, if there is an email coming to smith@company.com, then HipLink will send the body of the message to the receiver smith. Spaces are not allowed (as for all email addresses in general), thus it will not be possible to send messages to such receivers, even if HipLink allows Receivers to be defined with spaces in the name, e.g., Joe Smith.

Two-Way Email Address Configuration:

The SMTP Server Email Gateway allows defining an additional email address that will handle two-way requests. Messages sent to this email address trigger Response Actions. The output of a Response Action (if any) will be delivered to the sender.

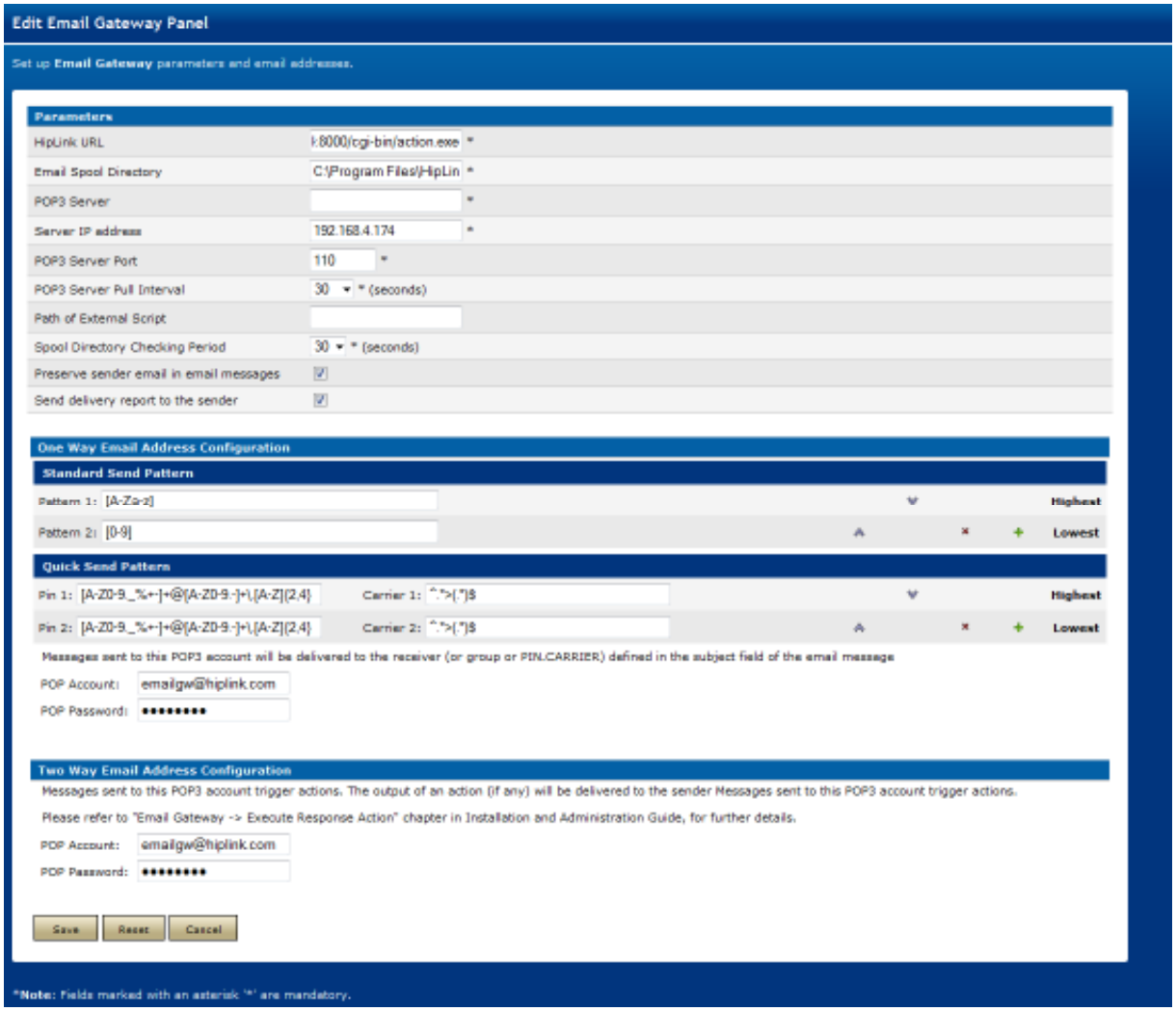
POP3 Email Gateway Parameters

To set the POP3 Server Email Gateway parameters:

- From the Settings menu, click Email Gateway on the left navigation bar.

©HipLink Software 2013. All Rights Reserved.

- On the Email Gateway Panel, click the POP3 radio button.
- Click the Edit button to reach the Edit Email Gateway Panel.



Edit Email Gateway Panel - POP3 Server

The **POP3 Server Email Gateway** parameters that can be set are:

- **HipLink URL** - specifies the URL of the HipLink instance used by the SMTP Server Email Gate-way to send messages. If there is only one HipLink installed on the machine, this is the default URL.
- **Email Spool Directory** - specifies the directory where the emails received by the gateway are stored.
- **POP3 Server** - specifies the IP address of the machine hosting the POP3 Email Gateway.
- **Server IP Address** - specifies the IP address of the machine that hosts the POP3 Server Email Gateway.
- **POP3 Server Port** (mandatory) - specifies the port number of the POP3 Server Email Gateway.
- Select the POP3 Server Pull Interval from the drop-down list (mandatory). The default value is 30 seconds.

©HipLink Software 2013. All Rights Reserved.

- **Path of External Script** - specifies the path of the external script to be executed when a message is sent by Email Gateway.
- **Spool Directory Checking Period** - specifies the time period at which the POP3 Server Email Gateway is checking the spool for new messages.
- **Preserve sender email in email messages** - the email address received by the recipient would contain email address of the original sender.
- **Send delivery report to the sender** - if checked, the delivery report would be sent back to the sender.

Enter the one-way messaging POP Account and POP Password in the One-Way Email Address Configuration section.

Enter the two-way messaging POP Account and POP Password in the Two-Way Email Address Configuration section.

Click the Save button to submit your changes and return to the Email Gateway Panel, Reset button to fill in the previous values, or Cancel button to return without saving.

Note: Any changes to the Email Gateway settings require the Email Gateway Service to be restarted.

- One-Way Email Address Configuration:
- Standard Send Pattern:
- Define a pattern(s) for the receiver name to send a standard message.
- Quick Send Pattern:
- Define a pattern(s) for the receiver pin and carrier to send a quick message.

Messages sent to the generic one-way email address are delivered to the recipient (Receiver or Group) which is defined in the Subject field of the email message. The Receiver or Group should be already defined in the HipLink database. For example, if there is an email sent to HipLink@mail.company.com with the subject Joe Smith, then HipLink will send the body of the message to the receiver Joe Smith.

Two-Way Email Address Configuration:

The Email Gateway allows defining an additional two-way email address that will handle two-way requests. Messages sent to this email address trigger Response Actions. The output of a Response Action (if any) will be delivered to the sender.

©HipLink Software 2013. All Rights Reserved.

SNPP Gateway

Introduction

The SNPP Gateway feature in HipLink is a front end interface for accepting incoming messages over the SNPP protocol and then sending outgoing messages to various devices using HipLink. Through this gateway, the HipLink product has the ability to listen to SNPP messages on defined TCP/IP port, process them & generate messages, and finally send them to existing receivers and / or groups, possibly belonging to a group of different protocols.

The HipLink SNPP gateway can communicate with any external SNPP message generating source, like certain CAD systems, provided they are connected through the same TCP/IP network. As a possibility, it can also communicate over the Internet if the HipLink server is configured to be accessed through a public IP, either directly or indirectly through port forwarding.

Note: Since HipLink has the capability of generating SNPP messages as well, the outgoing message can be fed into the SNPP Gateway of other HipLink servers as well.

The SNPP Gateway is using the SNPP (Simple Network Paging Protocol) protocol and allows HipLink to handle high volumes of messages more efficiently and in a very flexible and convenient way. All messages sent through the SNPP Gateway are Standard or Quick messages, even though HipLink offers a variety of sending functions via the GUI: Standard Send, Two-Way Send, Scheduled Send, or Custom Escalation Send.

SNPP Gateway Administration

The HipLink SNPP Gateway is enabled through your License Key. Please check the License Key Panel in the Settings tab to see if this feature is enabled for your version of HipLink. If it is not enabled, and you would like to use the SNPP Gateway, please contact HipLink Support at support@hiplink.com for further assistance.

SNPP Gateway Service

The SNPP Gateway service is similar to the other HipLink services. The Services tab allows the User to manually start or stop the SNPP Gateway service. This will be necessary if the SNPP Gateway parameters are changed.

SNPP Gateway Log File

The SNPP Gateway Log is a file that tracks SNPP Gateway operations.

To view the SNPP Gateway log file:

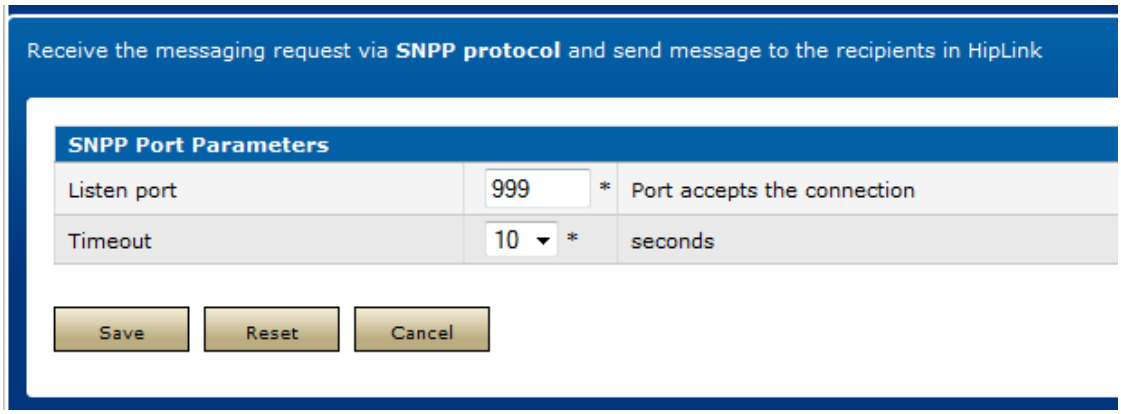
- In the Logs tab, click on the SNPP Gateway link in the left navigation bar.
- View log entries (The newest entries are displayed at the top of the page).
- Click Logs tab to return to the main Logs panel.

©HipLink Software 2013. All Rights Reserved.

SNPP Gateway User

For HipLink, the SNPP Gateway acts like a CLI client. For this reason, a CLI User account has to be created for the SNPP Gateway (i.e., a User of type CLI user or both that has an IP address - here the IP address of the machine where the HipLink server is installed). See the Users Panel section in the HipLink Administration chapter for details about creating a User.

SNPP Gateway Configuration



SNPP Gateway

Listen Port: The TCP port on which the SNPP Gateway Server will be listening for incoming connections.

Timeout: The inactivity timeout at any step during the SNPP protocol workflow after a connection has been established with an incoming connection and the processing of the protocol has started. If nothing is received within the time defined in this field, the SNPP Gateway assumes some error has occurred, stops further processing and drops the connection. This should be configured as per the network latency in the Production environment. For very fast networks, a minimal value should suffice, while for slower networks, a larger value is recommended.

SNPP Gateway Workflow

An SNPP message is generated by an external source, with the IP address and listening port of the SNPP Gateway defined as SNPP server address.

The SNPP message is generated from a location where the SNPP Gateway can be reached over the TCP/IP network.

The SNPP pager creates a socket connection with the SNPP Gateway provided the SNPP Gateway service is running and listening on the correct TCP/IP port.

The message received will consist of two parts: the receiver PIN and the message BODY.

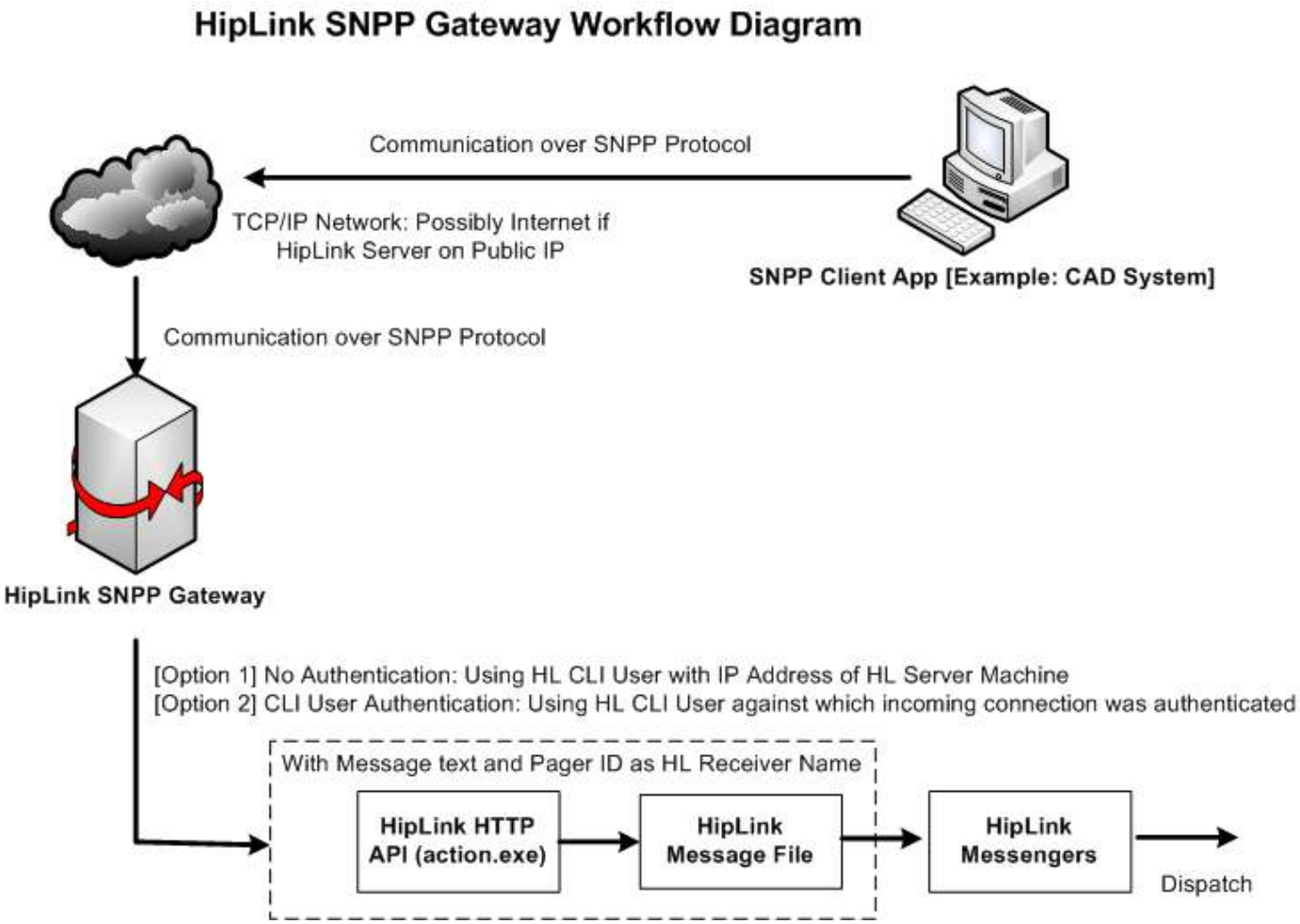
The SNPP Gateway will check for any receiver (or group) present in HipLink with the name matching with the received receiverPIN. (Note: Instead of providing the name or the receiver or group, the format of PIN. CARRIER can also be used where the CARRIER name has to be valid.)

©HipLink Software 2013. All Rights Reserved.

If a receiver is found in HipLink, the SNPP Gateway uses the HipLink HTTP API to create a HipLink message file for that receiver (that would picked up by a HipLink messenger and the message sent out to the HipLink Receiver's PIN).

If the receiver is not found, an error is returned to the SNPP message generating device/system.

The following diagram illustrates the workflow of SNPP Gateway:



SNPP Gateway Workflow

©HipLink Software 2013. All Rights Reserved.

TAP Gateway

Introduction

The TAP Gateway feature in HipLink is a front end interface for accepting incoming messages over the TAP protocol and then sending outgoing messages to various devices using HipLink. Through this gateway, the HipLink product has the ability to listen to TAP messages on connected serial ports, process them & generate messages, and finally send them to existing receivers and / or groups, possibly belonging to a group of different protocols.

The HipLink TAP gateway can communicate with any external TAP message generating source, typically TAP paging terminals, like the Zetron paging terminal, either connected physically over a serial cable (Leased mode), or through the telephone network using modems (Dial-up mode).

Note: Since HipLink has the capability of generating TAP messages as well, the outgoing message can be fed into the TAP Gateway of other HipLink servers as well, either using TAP Dial-up or TAP Leased mode.

The TAP Gateway is using the TAP (Telocator Alphanumeric Protocol) protocol and allows HipLink to handle high volumes of messages more efficiently and in a very flexible and convenient way, by supporting multiple simultaneous TAP configurations, and thus supporting handling parallel requests on each configured connection. For the sake of simplicity, all messages sent through the TAP Gateway are Standard or Quick messages, even though HipLink offers a variety of sending functions via the GUI: Standard Send, Two-Way Send, Scheduled Send, or Custom Escalation Send.

There are two different modes of operation for the TAP Gateway:

- **Leased Mode:** where the HipLink server and the TAP paging terminal are physically connected over a serial cable
- **TAP Dial-Up Mode:** where the HipLink server and the TAP Paging Terminal are not physically in the same location, and would communicate and acquire the connection between each other over the telephone network through modems.

TAP Gateway Administration

The HipLink TAP Gateway is enabled through your License Key. Please check the License Key Panel in the Settings tab to see that this feature is enabled for your version of HipLink. If it is not enabled, and you would like to use the TAP Gateway, please contact HipLink Support at support@hiplink.com for further assistance.

TAP Gateway Service

The TAP Gateway service is similar to the other HipLink services. The Services tab allows the User to manually start or stop the SNPP Gateway service. This will be necessary if the parameters defined in any configuration port in the TAP Gateway are changed.

TAP Gateway Log File

The TAP Gateway Log is a file that tracks TAP Gateway operations.

©HipLink Software 2013. All Rights Reserved.

To view the TAP Gateway log file:

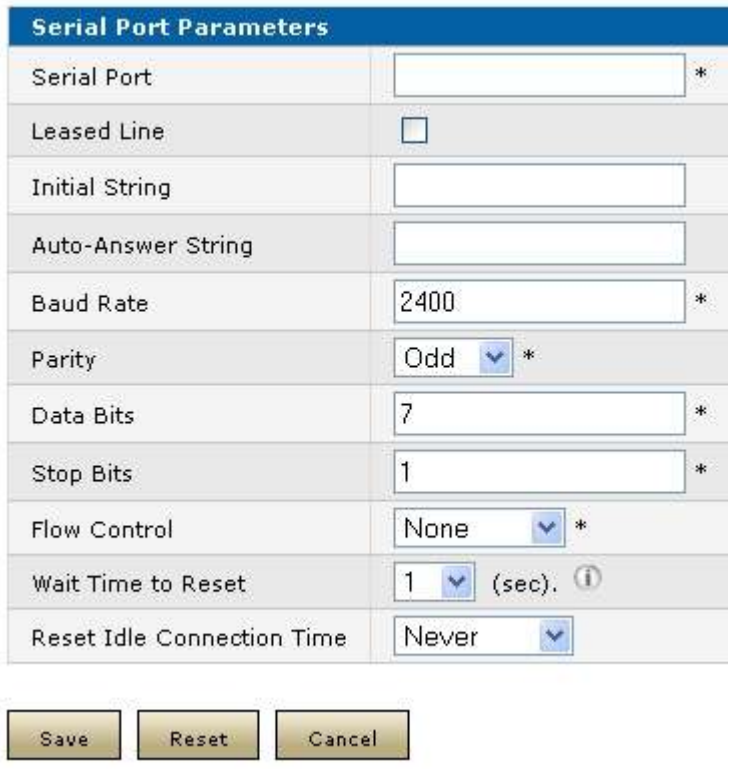
- In the Logs tab, click on the TAP Gateway link in the left navigation bar.
- View message entries. Note that the newest entries are displayed at the top of the page.
- Click Logs tab to return to the main Logs panel.

TAP Gateway User

For HipLink, the TAP Gateway acts like a CLI client. For this reason, a CLI User account has to be created for the TAP Gateway (i.e., a User of type CLI user or both that has an IP address - here the IP address of the machine where the HipLink server is installed). See the Users Panel section in the HipLink Administration chapter for details about creating a User.

HipLink TAP Gateway Configuration

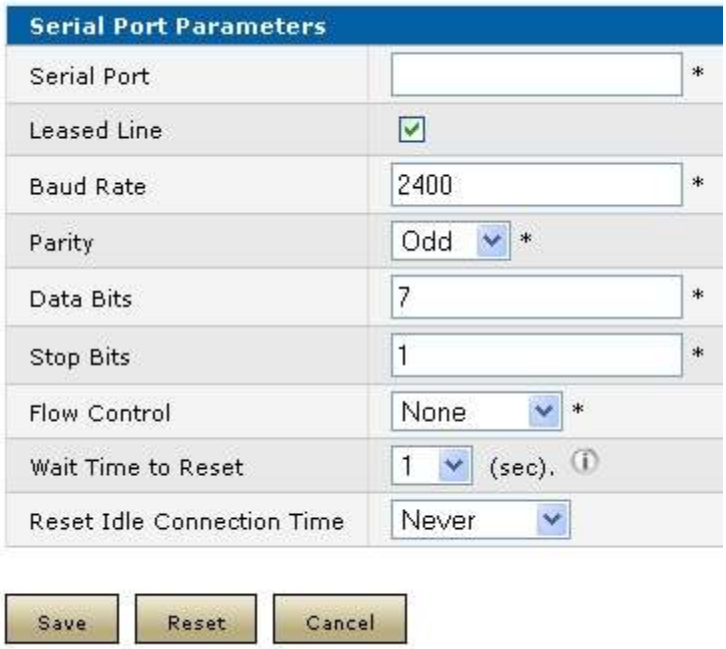
Dial-Up Mode



TAP Gateway – Dial-up Mode

TAP Leased

©HipLink Software 2013. All Rights Reserved.



TAP Gateway – Leased Mode

As could be seen in the screenshots, the Leased Line checkbox needs to be checked when configuring TAP Gateway in leased mode. This removes the fields Initial String & Auto-Answer String from the settings as they are not required in Leased mode. The rest of the fields are identical to those in Dial-Up mode.

Each of the fields is briefly described as follows.

- **Serial Port:** COM port in the system over which modem is connected
- **Leased Line:** Checkbox whether incoming TAP messages are expected through mode, or the TAP paging terminal is connected with a leased line
- **Initial String** *(Only available in Dial-up mode):* This is the initialization string of the modem. Most modems would work with "ATZ" but it could be typical to certain modems. See the Help window for further details on this.
- **Auto-Answer String** *(Only available in Dial-up mode):* This is the auto-answer string with which the modem is to be initialized. For Hayes-compatible modems, this would typically be setting up the value of the S0 register. In a typical scenario, where the modem should answer any incoming connections after the first ring, it should contain "ATS0=1".
- **Baud Rate:** As compatible with the modem hardware connected to the Serial Port
- **Parity:** As compatible with the modem hardware connected to the Serial Port
- **Data Bits:** As compatible with the modem hardware connected to the Serial Port
- **Stop Bits:** As compatible with the modem hardware connected to the Serial Port
- **Flow Control:** As compatible with the modem hardware connected to the Serial Port
- **Wait time to reset:** Certain modems (or hardware connected to the serial port) would need time to reset before being initialized again, typically 1 or 2 seconds. This also needs to be as compatible with the hardware connected to the Serial Port
- **Reset Idle Connection Time:** Certain modems (or hardware connected to the serial port) go into a bad state if there's no communication over an extended period of time.

©HipLink Software 2013. All Rights Reserved.

This could be never required, so that's one option. Otherwise, after the defined period, HL Tap Gateway will refresh the connection by closing the connection with the serial port and opening it again.

TAP Gateway Workflow

Basic Common Workflow of TAP Gateway

- The TAP Gateway will constantly listen on the COM serial ports defined in each profile.
- The message received will consist of two parts: the receiver PIN and the message BODY.
- The TAP Gateway will check for any receiver (or group) present in HipLink with the name matching with the received receiver PIN. (Note: Instead of providing the name or the receiver or group, the format of PIN.CARRIER can also be used where the CARRIER name has to be valid.)
- If a receiver is found in HipLink, the TAP Gateway uses the HipLink HTTP API to create a HipLink message file for that receiver (that would be picked up by a HipLink messenger and the message sent out to the HipLink Receiver's PIN).
- If the receiver is not found, an error is returned to the TAP paging terminal.

Specific Workflow in Dial-Up Mode

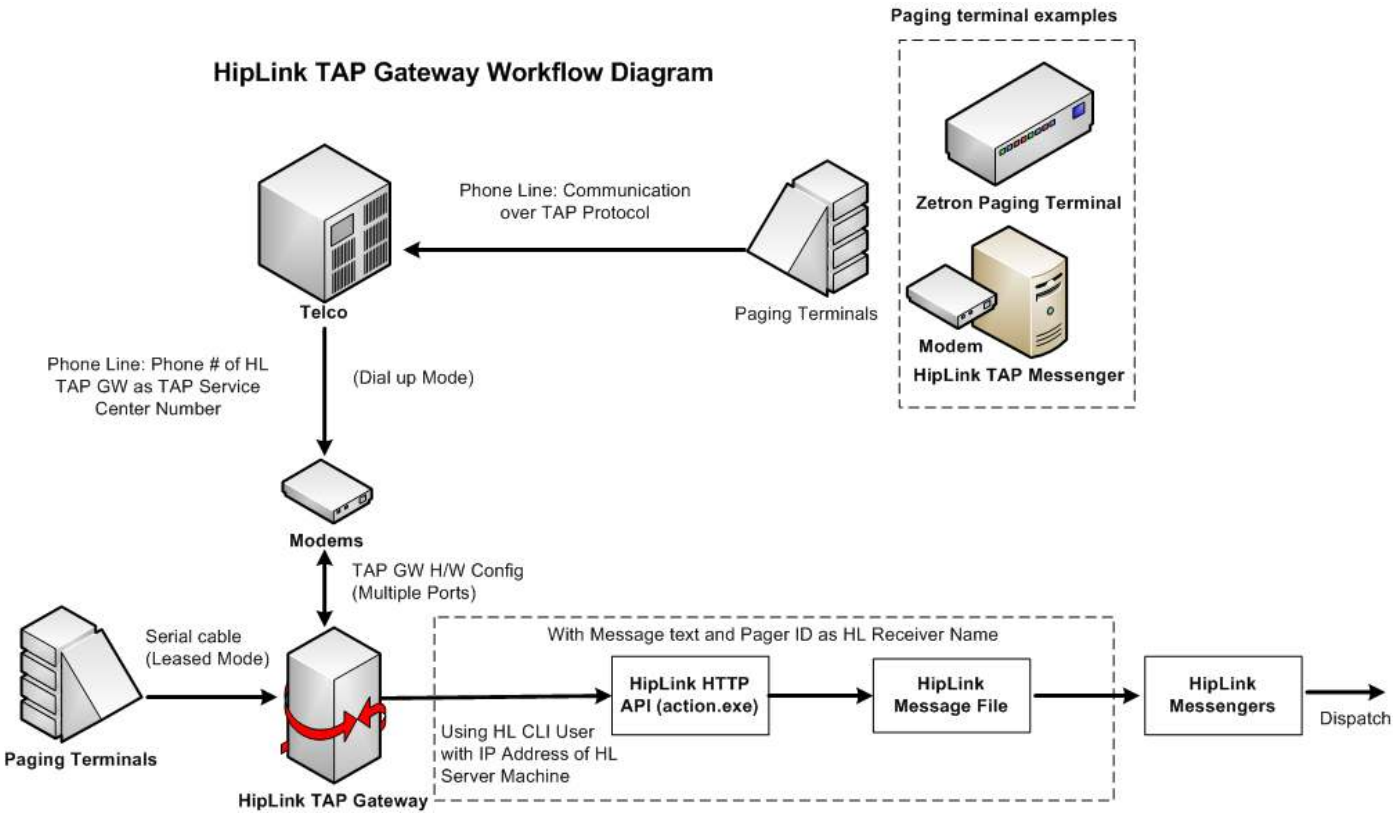
- In the Dial-Up mode, the message is generated through a TAP paging terminal (e.g. Zetron paging terminal) that is connected to a phone line using a dial-up modem and it communicates over TAP protocol through the telephone network.
- The paging terminal has the phone number of target TAP gateway defined as TAP service center number on which it dials using its modem.
- After successful handshake with the TAP Gateway's modem, the paging terminal transfers the receiver's pin and message to the TAP Gateway.
- The receiver's pin is an actual device number in the TAP Gateway's database over which the gateway forwards the message.
- If the pin is invalid, the gateway returns an error to the paging terminal.

Specific Workflow in Leased Mode

- In the Leased mode, the message is generated through a TAP paging terminal (e.g. Zetron paging terminal) that is connected directly to the HipLink Server machine through a serial cable connected to a COM port.
- The paging terminal only has to establish communication with the TAP Gateway over the connected COM port.
- After successful communication with the TAP Gateway over the COM port the paging terminal transfers the receiver's pin and message to the TAP Gateway.
- The receiver's pin is an actual device number in the TAP Gateway's database over which the gateway forwards the message.
- If the If the pin is invalid, the gateway returns an error to the paging terminal.

The following diagram illustrates the workflow of TAP Gateway:

©HipLink Software 2013. All Rights Reserved.



TAP Gateway Workflow

©HipLink Software 2013. All Rights Reserved.

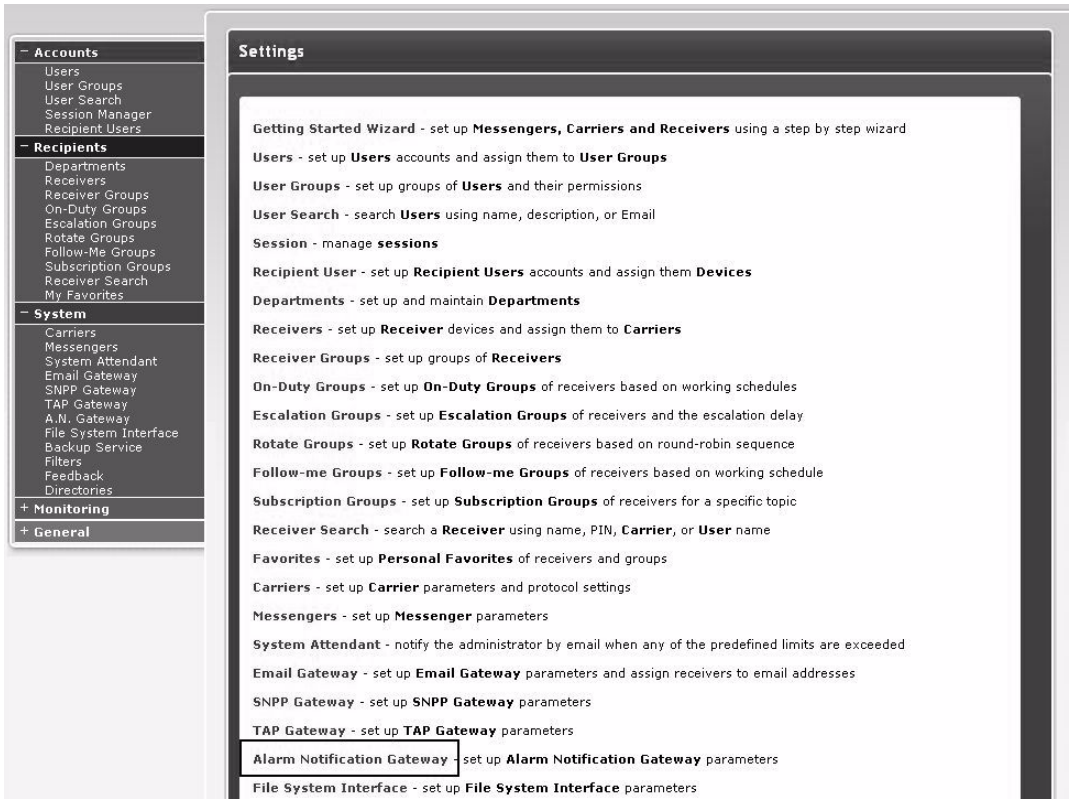
Alarm Notification Gateway

Alarm Notification Gateway is an optional feature that is available based on your HipLink license that is purchased. This feature lets the administrator set up the Alarm Notification Gateway profiles- a general serial connector that is capable of providing simultaneous connectivity of several serial devices through different serial port configurations to HipLink. This expands HipLink's support of various appliances and integration with several different solutions for emergency response, management and control. Alarm Notification Gateway:

- Provides regular expression parser capability to filter out needed text from the input stream.
- Allows the user to create a text string dynamically from the previously parsed input chunks.
- Can determine the target recipient name from the input stream, and sends the customized text string as message to the recipient.
- Allows outputting the text to some text file as special log entries.
- Allows the user to write back to the Serial port.
- Allows the user to perform data transformations and manipulations on the input data.

Alarm Notification Gateway Main Panel

The Alarm Notification Gateway (A.N. Gateway) panel can be accessed from Settings tab on the left navigation bar in Systems menu as well as from the main panel.

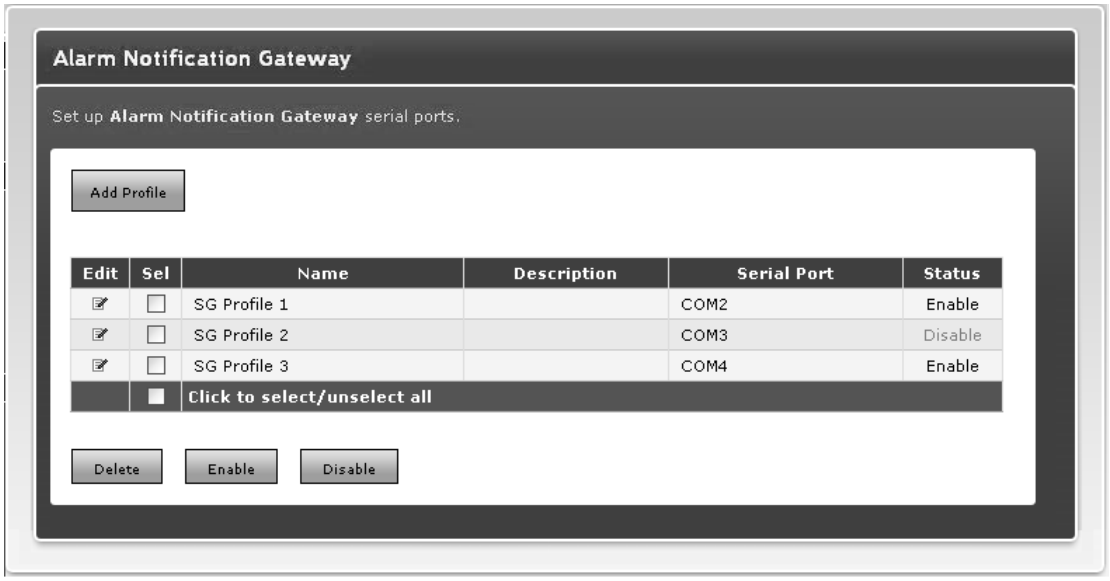


Alarm Notification Gateway link on Settings Panel

©HipLink Software 2013. All Rights Reserved.

This panel contains all the Alarm Notification Gateway profiles. It allows you to perform following operations:

- Add profile
- Edit profile
- Enable profile
- Disable profile
- Delete profile



Alarm Notification Gateway Main

- Add profile: Click on the Add Profile button to go to the Add Serial Port for the Gateway panel.
- Edit profile: Click on the Edit icon next to any existing profile to go to the Edit Serial Port for the Gateway panel and modify its settings.
- Enable profile: Select one or more disabled profiles and click the Enable button to enable them.
- Disable profile: Select one or more enabled profiles and click the Disable button to disable them.
- Delete profile: Select one or more profiles and click the Delete button.

Note: Enabling, disabling or deleting profiles requires Alarm Notification Gateway service to be restarted.

Add/ Edit Alarm Notification Gateway Profile

The following screen allows a user to create a new profile or edit an existing one. The fields marked with an * are mandatory.

Profile Parameters

- **Name:** Create a unique name for each profile.

©HipLink Software 2013. All Rights Reserved.

- **Description:** Enter a brief description of the profile.
- **Enable Profile Logs:** Select this check box to generate logs for the particular profile.

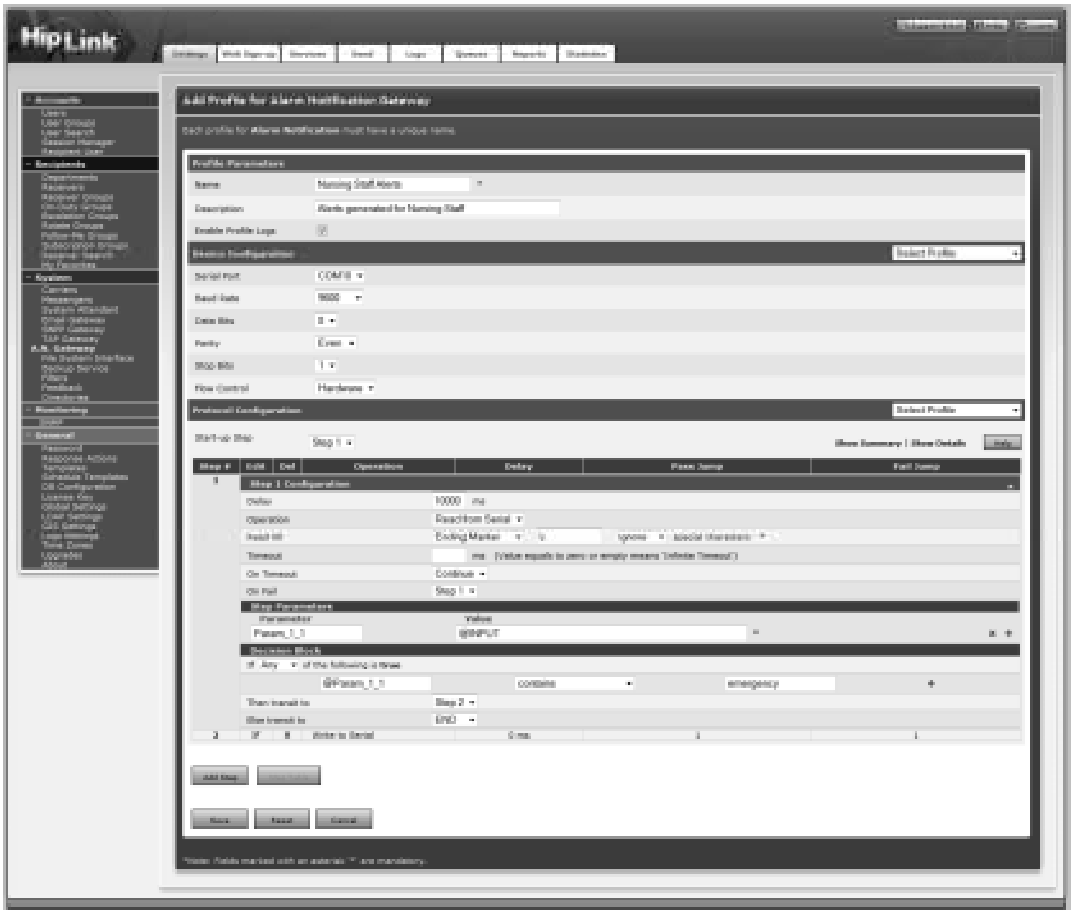
Device Configuration

This section contains configurations required to define the connection to the serial device. It consists of following drop down lists:

- **Serial Port:** Select serial port to be used by the profile.
- **Baud Rate:** Select baud rate to be used by this profile
- **Data Bits:** Select number of data bits to transmit at a time.
- **Parity:** Select the type of parity checking.
- **Stop Bits:** Select number of bits used to indicate the end of data transmission.
- **Flow Control:** Select data flow mechanism between the devices.

Serial Port should be unique for each profile. All other configurations (Baud Rate, Data Bits, Parity, Stop Bits, and Flow Control) should match those used for the serial device in use. Alarm notification Gateway provides the ability to copy and replicate a profile's configuration in another profile.

A combo box on the right side of Device Configuration section contains the list of all the existing profiles. Selecting a profile from the list fills in all the fields of Device Configuration section with the values saved in the selected profile.



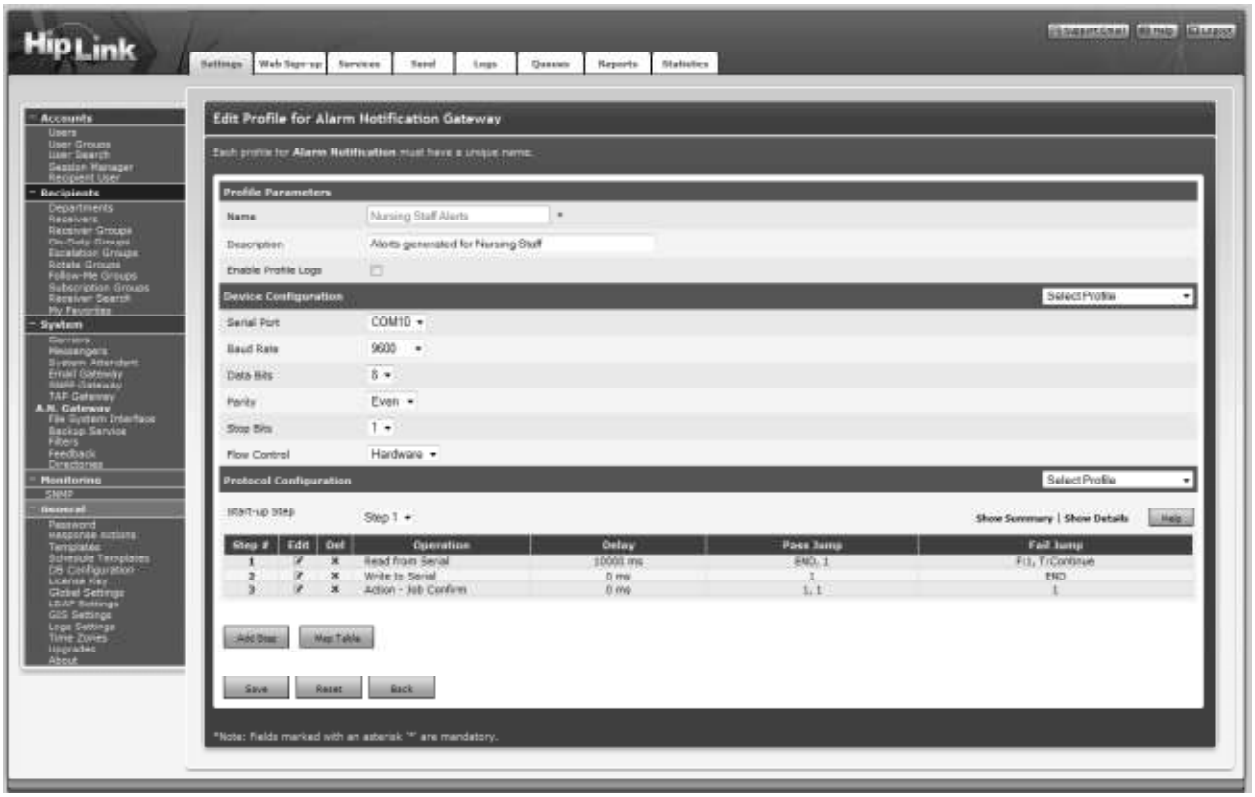
©HipLink Software 2013. All Rights Reserved.

Protocol Configuration

This section defines which protocol engine will be executed, and its configuration parameters.

- **Start-up Step:** To decide which step would be executed first from the defined steps. Unless there is a step added in the profile, this field will remain disabled.
- **Add Step:** Allows adding an additional step to the profile. It is explained more in the next section.
- **Map Table:** Opens up Data Map Table pop-up window. This button becomes enabled when the profile is saved for the first time. Map Table is described later in detail.
- **Help:** Opens up Parameters and Functions Table pop-up window. This is described later in detail.

A combo box on the right side of Protocol Configuration section allows copying and replicating an existing profile's configuration in your profile. Selecting a profile from the list fills in all the fields of Protocol Configuration section with the values saved in the selected profile.



Alarm Notification Gateway - Protocol Configuration.

Adding Steps to a Profile

When a serial profile is being added or edited, a new step can be added by pressing Add Step button. This will open Step Configuration form in the Protocol section of the profile.

• **Show Summary:** Clicking on Show Summary link on the right side of Protocol Configuration header collapses all the steps and shows their summary. When the profile is saved or when it is opened in edit mode, all the steps are collapsed i.e. they are shown in summary mode. To view the detail mode, click on the Show Details link. To view the details of a particular step, click the

©HipLink Software 2013. All Rights Reserved.

Edit icon next to that step. To view a summary of a particular step, click the Collapse button on the right side of that step.

When in summary mode, following actions could be performed on a step:

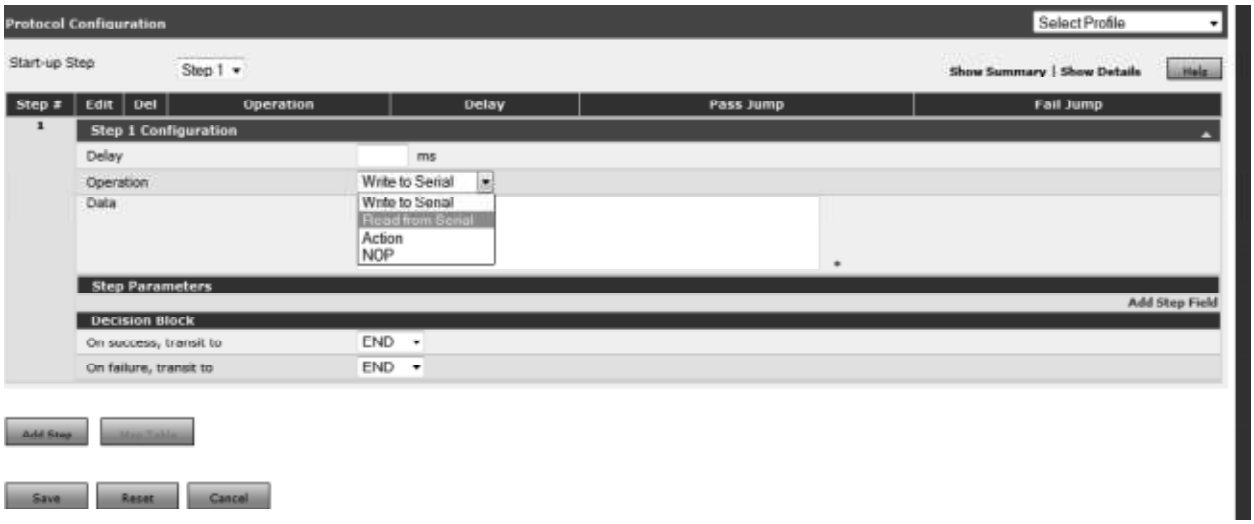
- **Edit:** Click on the edit link next to a step to expand it and to modify its contents.

- **Delete:** Click on the delete link next to a step to delete it.

Note: *A step cannot be deleted if it is being used as start-up step or as transition step in some other step's configuration.*

Operations

A step is further categorized into four types, and each one is associated with an operation. Based on the type of the operation, the form will change itself to include appropriate fields required for that operation.



Alarm Notification Gateway - Operations selection

Other than Write, all operations in a step support parameter extraction and decision making.

- Parameter extraction: extracting and processing data coming from serial port or Response Action.

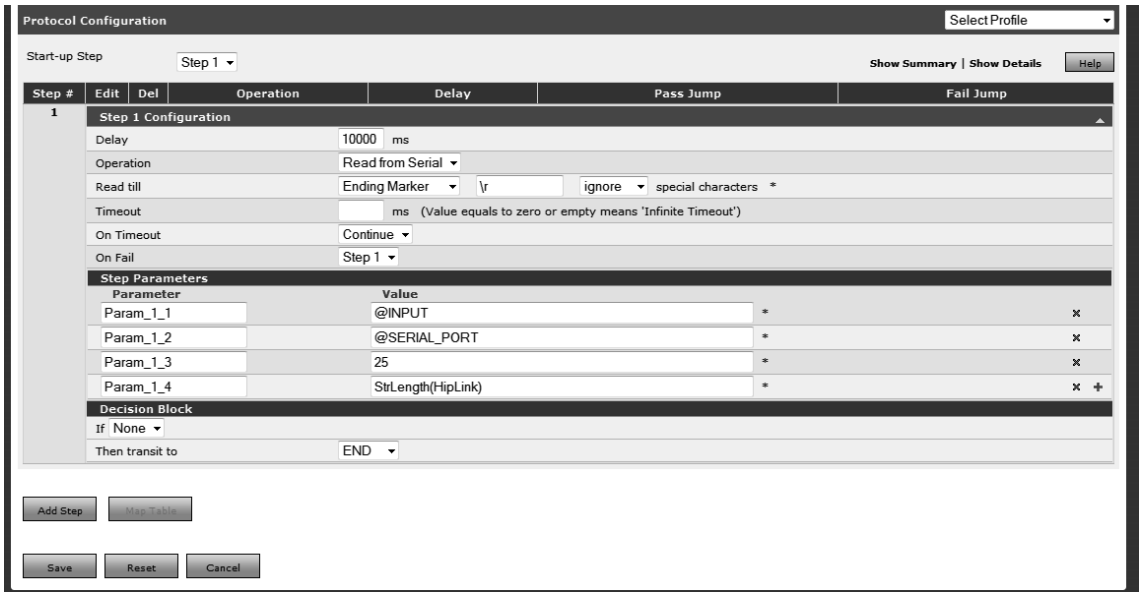
- Decision making: based on the value of Parameter the next step to follow is chosen.

Step Parameters

The values of step parameters (variables) are obtained by evaluating them. Alarm Notification Gateway supports many types of variable qualifiers that take input arguments and provide an output as result. The result of qualifiers can be assigned to parameters and two or more qualifiers can be used to concatenate their results to form a more sophisticated data string.

Note: Qualifiers are described later in detail.

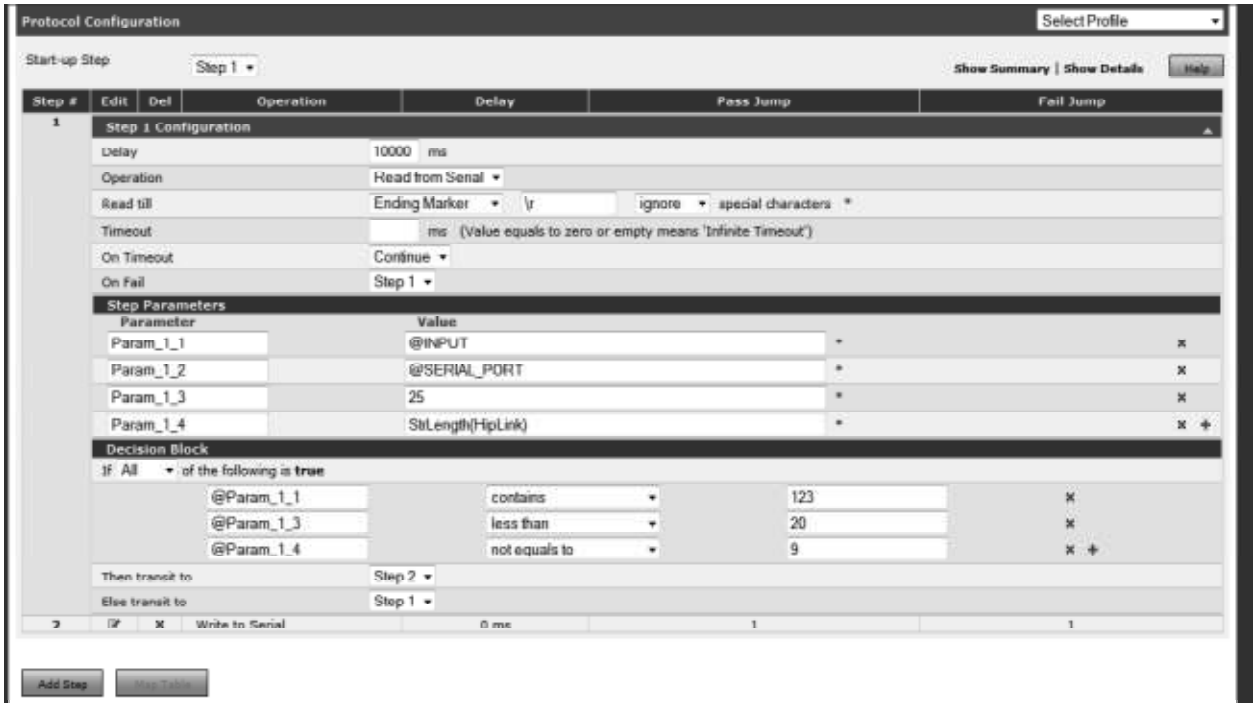
©HipLink Software 2013. All Rights Reserved.



Alarm Notification Gateway - Step Parameters

Decision Block

Decision Block is used to define transition to the next step based on the values of the parameters.



Alarm Notification Gateway - Decision Block

Once all parameters have been evaluated, an IF condition will be prepared. The IF condition is designed to accommodate three clauses over multiple parameters:

- All (Similar to logical AND) - Returns true If all given conditions are true.

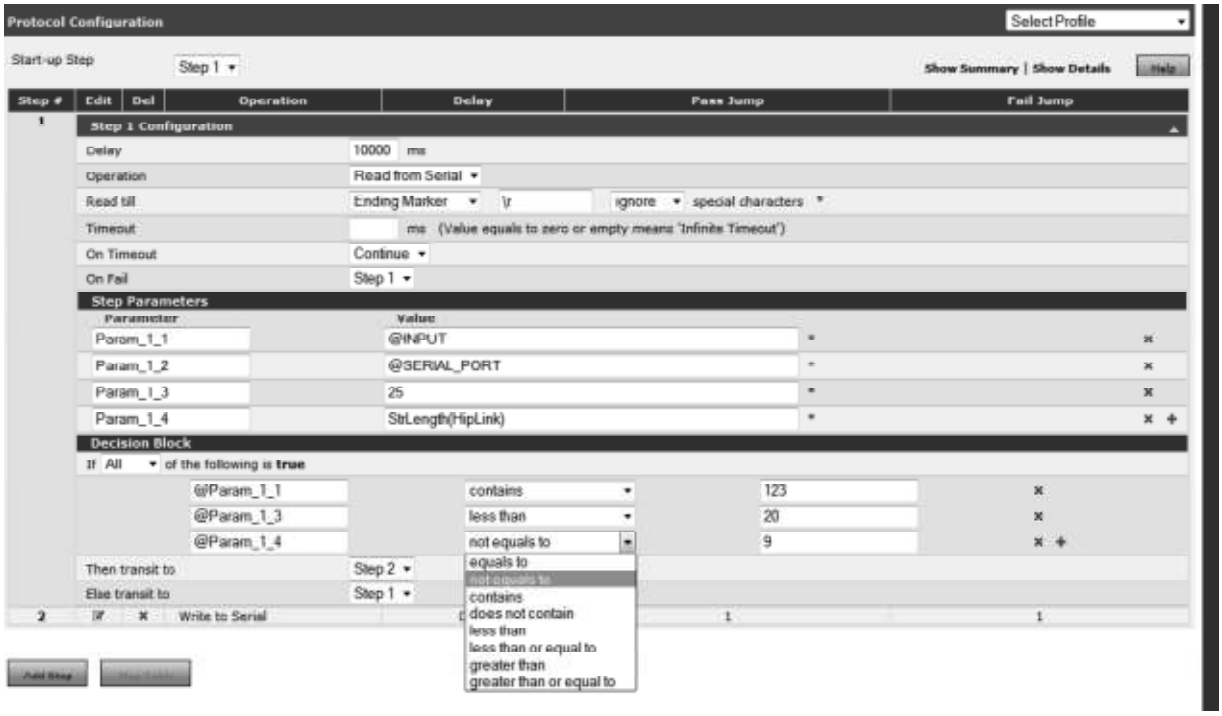
©HipLink Software 2013. All Rights Reserved.

- None (Similar to logical AND) - Returns true if none of the given conditions is true.
- Any (Similar to logical OR) - Returns true if any one of the given conditions is true.

The equality operators applied on parameters are of many types. These include:

- Equals to
- Not equals to
- Contains
- Does not contain
- Less than
- Less than or equal to
- Greater than
- Greater than or equal to

The condition can evaluate to a TRUE or FALSE depending upon the parameters' values.



Alarm Notification Gateway - Decision Block, Logical Operators

If the result of IF condition is true, the next step to follow is defined in Then transit to field. The step to follow on the failure of IF condition is specified in Else transit to field.

There are two types of transitions allowed in both Then transit to and Else transit to fields:

- **END:** To end profile execution.
- **Step X:** To transit to a particular step.

©HipLink Software 2013. All Rights Reserved.

Operations

Protocol state machine is configured by adding steps. A step is further categorized into four types, and each one is associated with an operation. Hence there are four types of operations:

- **Read:** This operation reads data from serial port.
- **Write:** This operation writes data to serial port.
- **NOP:** This operation enables the user to define dumb step for data transformation or translation.
- **Action:** This operation executes HipLink's Response Action.

Read

Read operation instructs the Alarm Notification Gateway to read from the serial port. It supports parameter extraction; this means certain data can be extracted from the incoming data.

Alarm Notification Gateway - Read Operation

Parameters defined for this operation are:

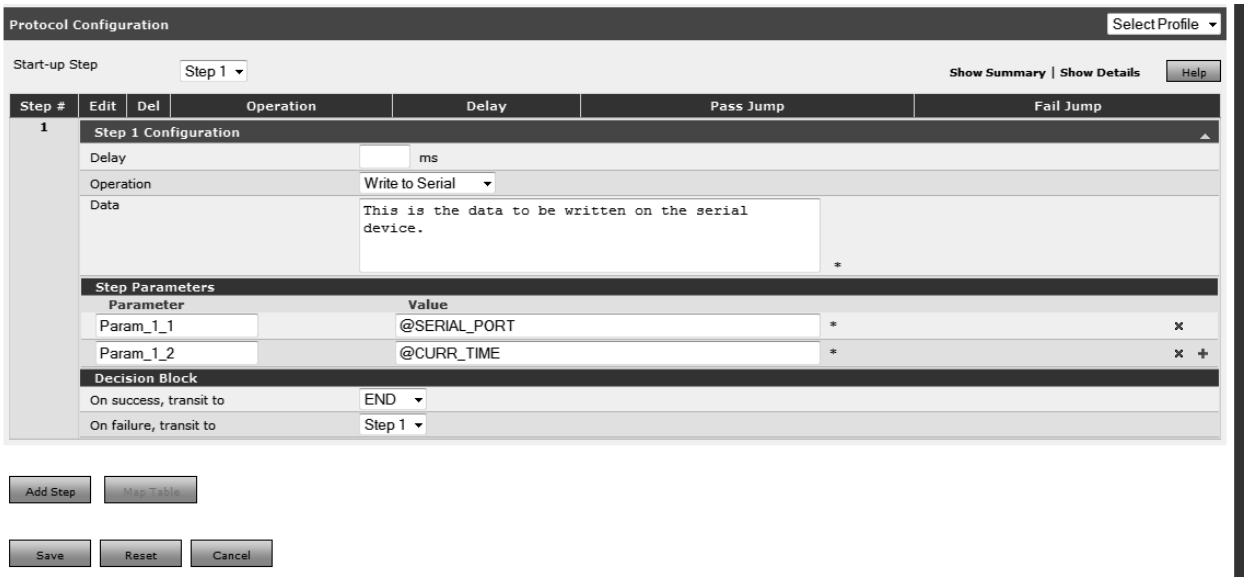
- **Delay:** This field is common in all operations. Setting up a delay instructs Alarm Notification Gateway to have a pause before processing the operation. Its default value is zero, and its unit is millisecond (ms).
- **Ignore / Encode special characters:** If ignore is selected, special characters read from the serial device will be ignored else they would be encoded.
- **Read Till:** This defines the point up till which data will be read from the serial port. It is a drop down list which contains two values:
- **Ending Marker:** Requires the user to define the delimiter character in the field.
- **Maximum Length:** Requires the user to define the maximum number of characters to be read at a time.

©HipLink Software 2013. All Rights Reserved.

- **Timeout:** It defines how much time (in ms) to wait on Read operation before timing out. A zero (0) or null value means indefinite wait.
- **On Timeout:** If a read operation is timed-out, user can decide whether to move on to another step, continue the current step, or end profile execution.
- **Step X:** Move to the selected step on time out.
- **END:** End profile execution on time out.
- **Continue:** Continue the current step (loop back) on time out.
- **On Fail:** If a read operation fails due to bad connection, user can decide whether to move on to another step, or end profile execution.
- **Step X:** Move to the selected step on time out.
- **END:** End profile execution on time out.

Write

Write operation writes data to the serial port.



Alarm Notification Gateway - Write Operation

Parameters defined for this operation are:

- **Delay:** Setting up a delay instructs Alarm Notification Gateway to have a pause before processing the operation. Its default value is zero, and its unit is millisecond (ms).
- **Data:** This contains data to be written on the serial device. The data field can have both static data as well as previously extracted parameters (mandatory).
- **Decision Block:** Transition to next step can be defined in this block on write operation failure and success.

If Write operation is successful, the next step to follow is defined in On success, transit to field.

The step to follow on the failure of Write operation is specified in On failure, transit to field.

There are two types of transitions allowed in both On success, transit to and On failure, transit to fields:

- **END:** To end profile execution.

©HipLink Software 2013. All Rights Reserved.

- **Step X:** To transit to a particular step.

Action

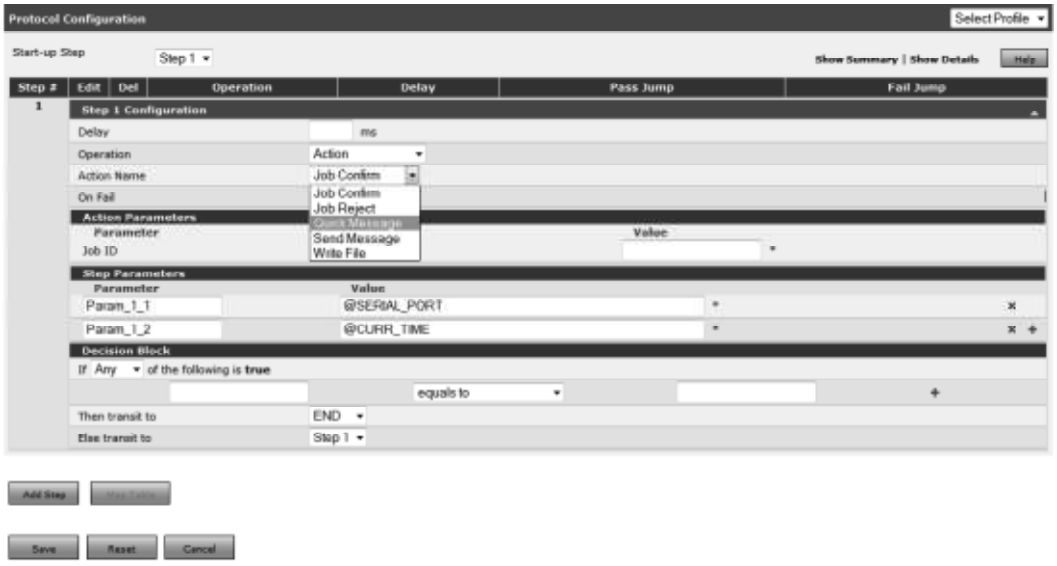
Action step can execute any Response Action defined in HipLink along with some specialized actions such as writing text to file(s).

Selecting a Response Action will list the parameter needed for that particular Response Action. The values to these parameters can be constants or values from the previously defined parameter variables.

The Action step supports the parameter extraction from the Response Action output, and conditional transitions can be defined based on their values.

The following actions are supported by this operation:

- Job Confirm
- Job Reject
- Quick Message
- Send Message
- Write File

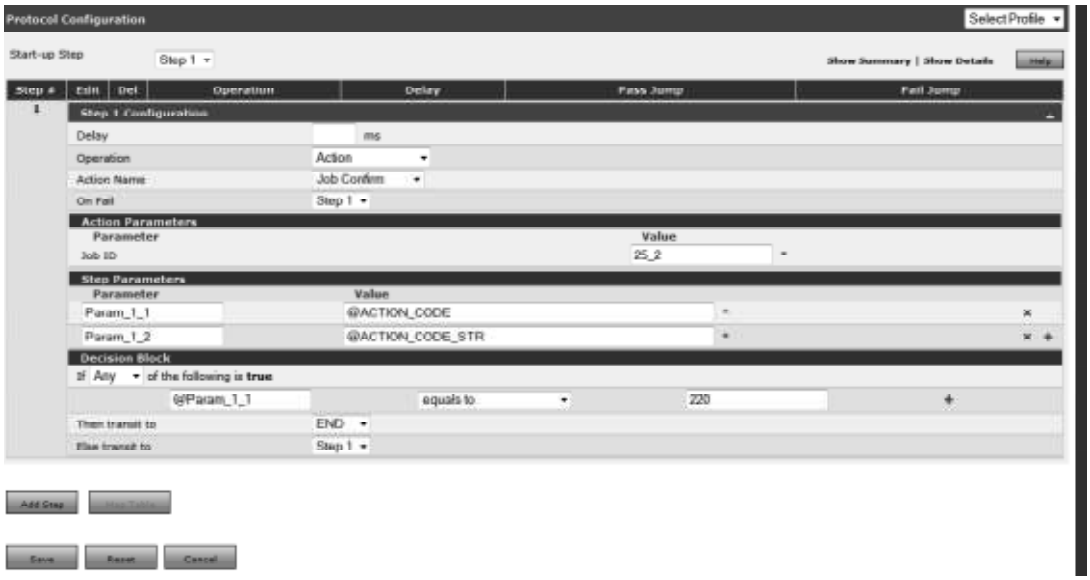


Alarm Notification Gateway - Action Operation

JOB CONFIRM

This action confirms a job sent via HipLink.

©HipLink Software 2013. All Rights Reserved.



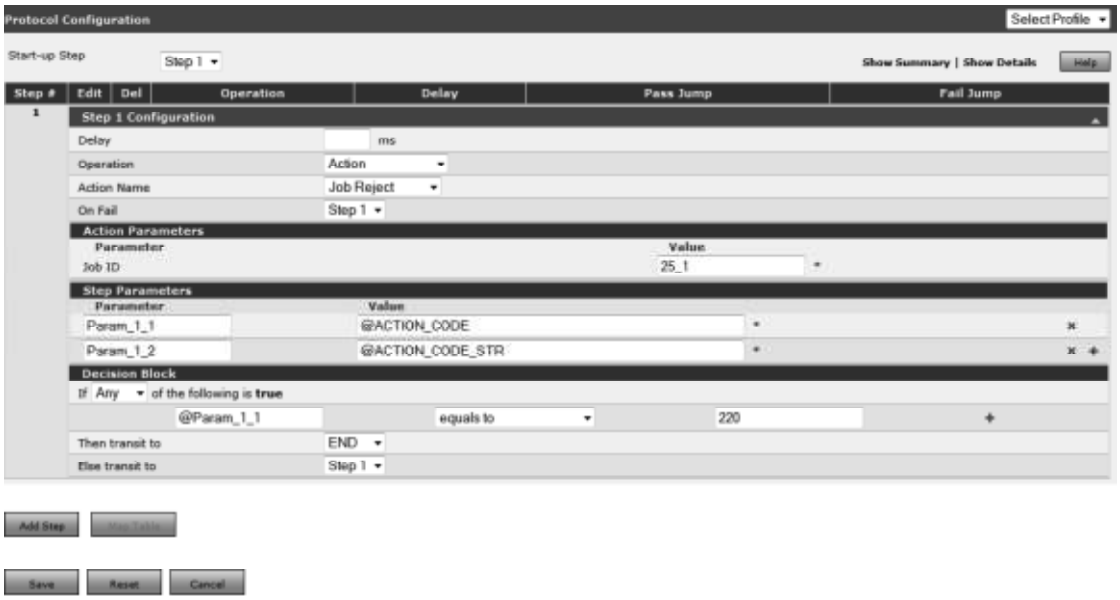
Alarm Notification Gateway - Action Operation, Job Confirm

Parameters defined for this action are:

Job ID: ID of the job to be confirmed. Can be a constant or value from a previously defined parameter variable (mandatory).

JOB REJECT

This action rejects a job sent via HipLink.



Alarm Notification Gateway - Action Operation, Job Reject

Parameters defined for this action are:

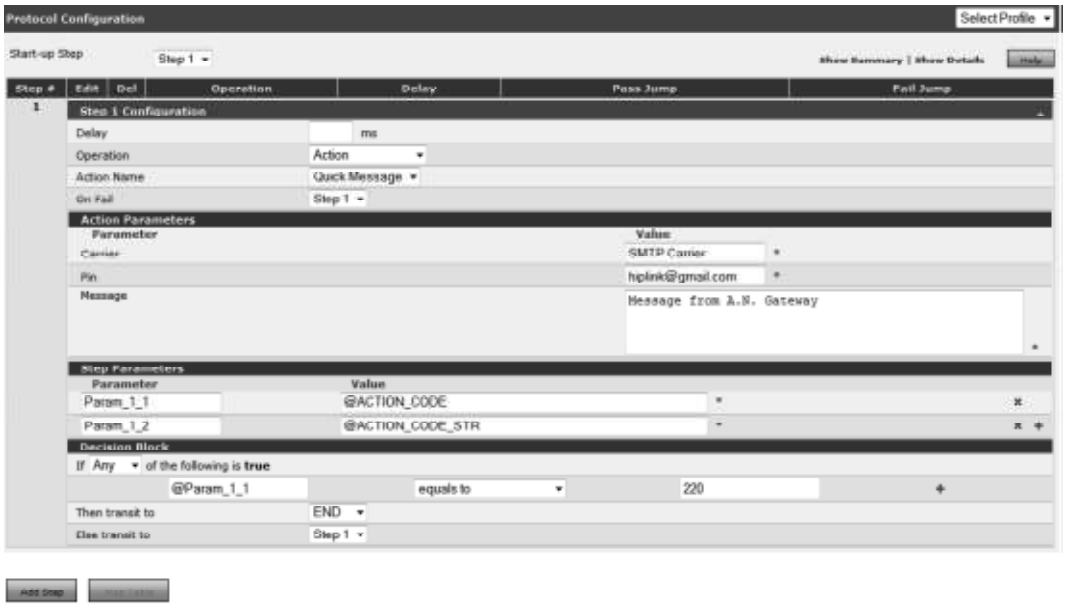
©HipLink Software 2013. All Rights Reserved.

Job ID: ID of the job to be rejected. Can be a constant or value from a previously defined parameter variable (mandatory).

Quick Message

This action is used to send quickly a message to a device that is not set up as receiver in HipLink. A message can be sent to any device using a pin number and an assigned carrier. Parameters defined for this action are:

- **Carrier:** A carrier can be used for message sending. This is the carrier defined in HipLink.
- **Pin:** A pin number of the device to which the message will be sent.
- **Message:** Body of the message.



Alarm Notification Gateway - Action Operation, Quick Message

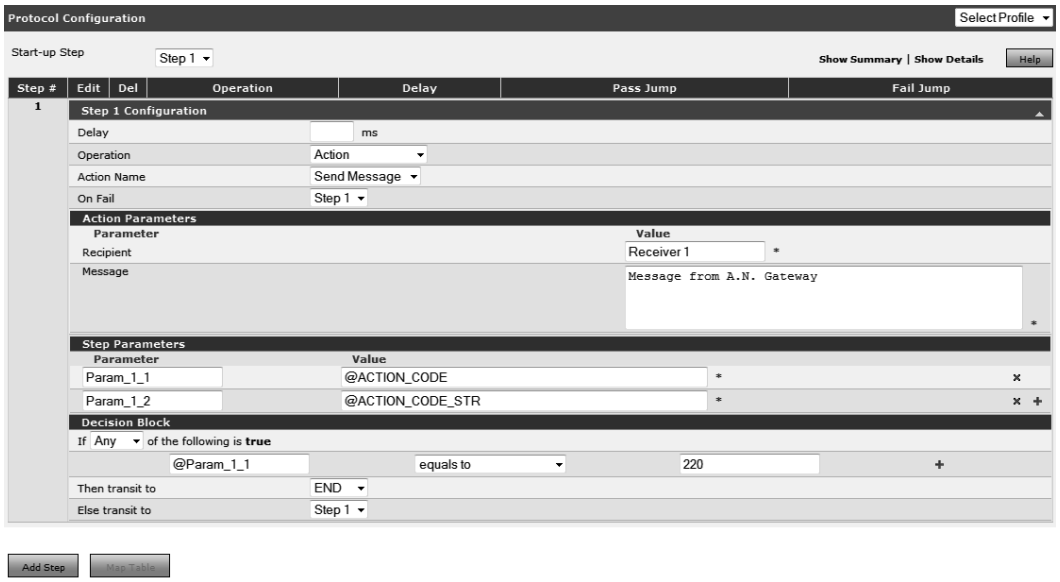
SEND MESSAGE

This action is used to send messages to receivers and receiver groups which have already been set up in HipLink. Parameters defined for this action are:

Recipient: Name of the receiver/ receiver group to whom the message will be sent.

Message: Body of the message.

©HipLink Software 2013. All Rights Reserved.



Alarm Notification Gateway - Action Operation, Send Message

WRITE FILE

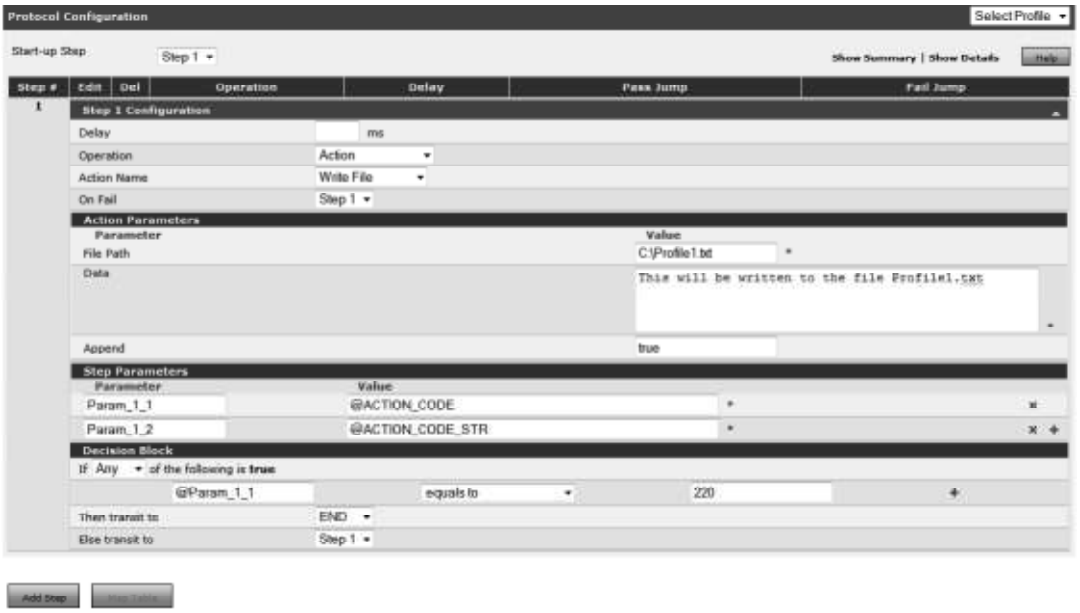
This action is used to write any text string to a file on the server. The purpose of this operation is messages to receivers and receiver groups which have already been set up in HipLink. Parameters defined for this action are:

File Path: Complete path of the file to which text string would be written. If the file does not exist, a new file will be created and data would be written to it. If the file exists, data would be appended/ overwritten depending upon the value in Append field.

Data: Data to be written to the file.

Append: Can have either true or false. If the file exists and Append field contains true, data would be appended to the contents of the file. If it contains false, contents of the file would be overwritten by the data provided.

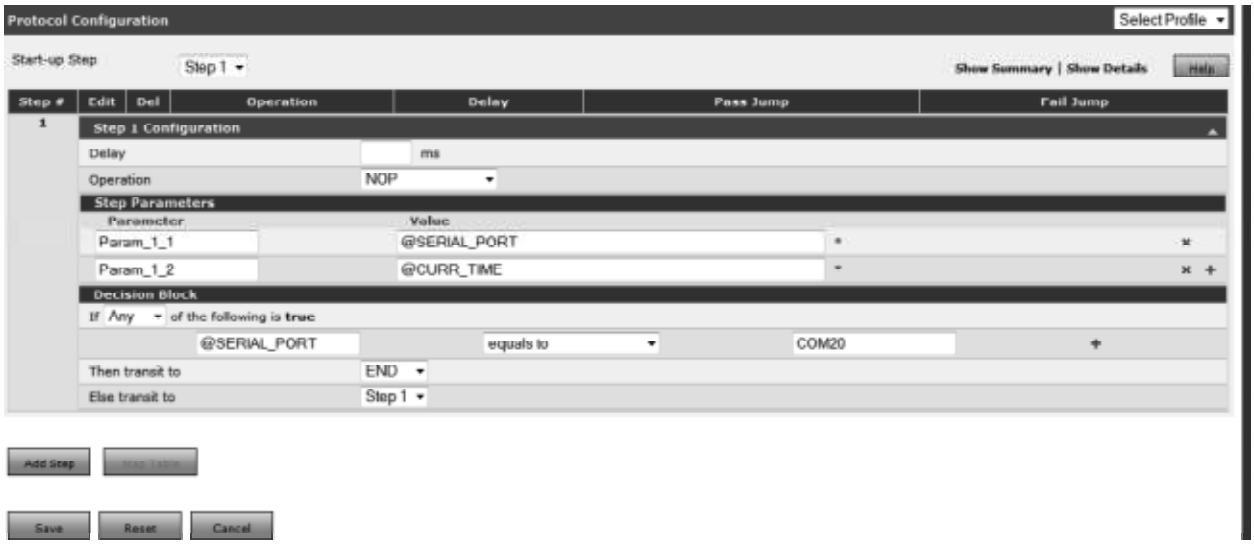
©HipLink Software 2013. All Rights Reserved.



Alarm Notification Gateway - Action Operation, Write File

NOP

A NOP does nothing on the serial interface; it can be thought of as an intermediate data processing step. More parameter variables can be prepared in this step, and conditional transitions can be taken on their values. It provides a mean to separate data transformation and translation steps in the state machine.



Alarm Notification Gateway - NOP Operation

©HipLink Software 2013. All Rights Reserved.

Parameters defined for this operation are:

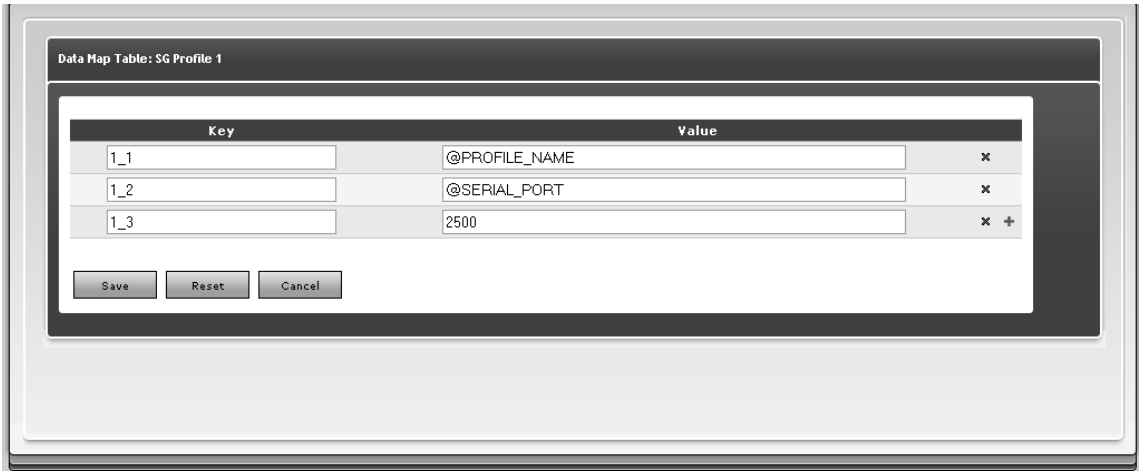
Delay: Setting up a delay instructs Alarm Notification Gateway to have a pause before processing the operation. Its default value is zero, and its unit is millisecond (ms).

Map Table

Map Table is a data repository for the users to save key-value pairs as data items. The user can save their data in the map table as values assigned to keys. At run-time, user can query the data using function GetFromMap(string key).

Map Table acts as a volatile memory i.e. user can change values at run-time using function SetToMap(string key, string value). These values persist for as long as the Alarm Notification Gateway service is running and the profile is being executed. The values can be changed permanently only by making changes in the Map Table

To access Map Table of a profile, click Map Table button on Edit Serial Port for Gateway panel. This will open Data Map Table pop-up for the particular profile.



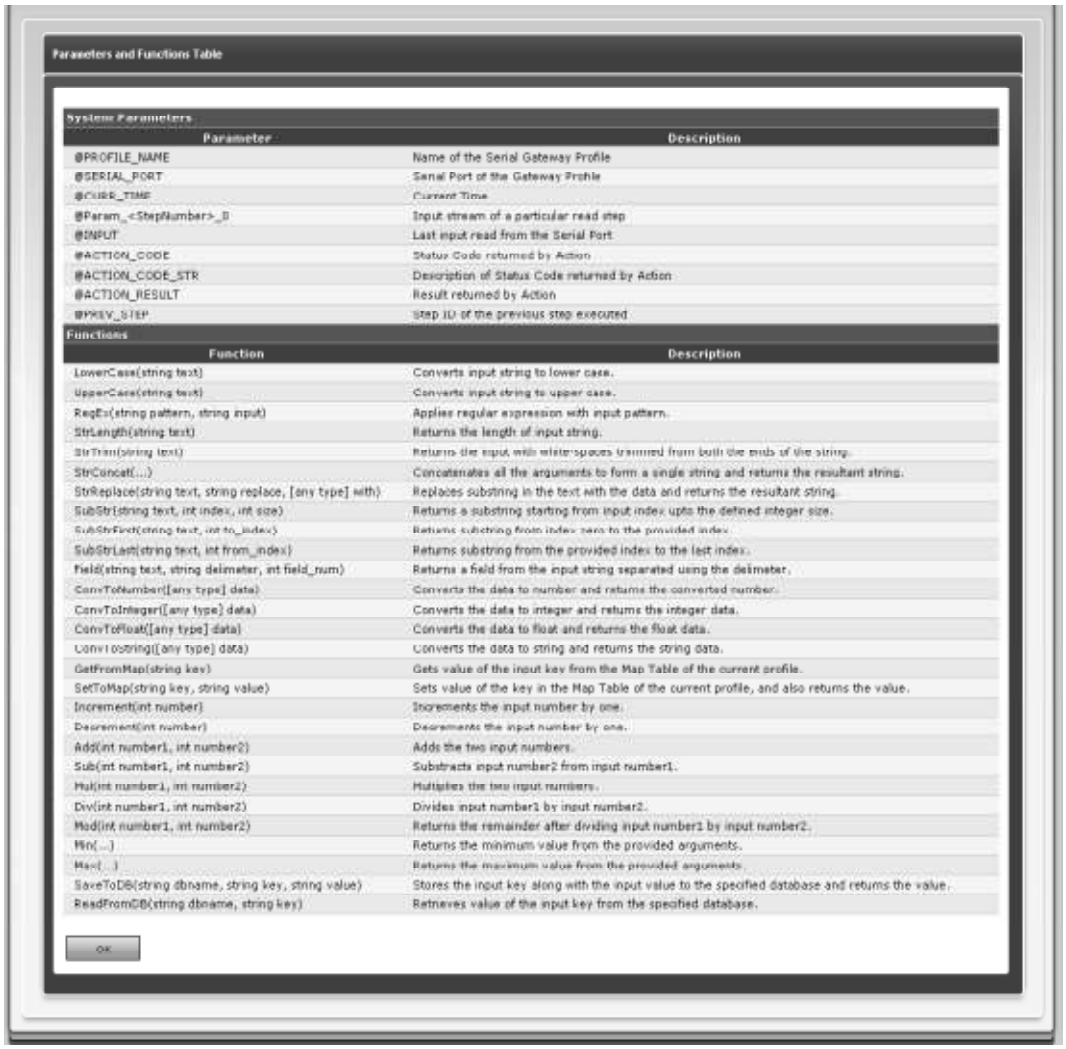
Alarm Notification Gateway - MAP Table

Help Menu

The help menu lists all the System Parameters and Functions used in the Alarm Notification Gateway profiles. Each parameter is followed by a brief description that explains its use in the serial profile. Each function is listed with its syntax followed by its brief description. To access Help menu, click on Help button on Edit Serial Port for Gateway panel. It will open Parameters and Functions Table pop-up window.

For details on System Parameters and Functions and their use, refer to Alarm Notification Gateway Help Document.

©HipLink Software 2013. All Rights Reserved.



Parameters and Functions Table

System Parameters

Parameter	Description
@PROFILE_NAME	Name of the Serial Gateway Profile
@SERIAL_PORT	Serial Port of the Gateway Profile
@CURR_TIME	Current Time
@Param_<StepNumber>_0	Input stream of a particular read step
@INPUT	Last input read from the Serial Port
@ACTION_CODE	Status Code returned by Action
@ACTION_CODE_STR	Description of Status Code returned by Action
@ACTION_RESULT	Result returned by Action
@PREV_STEP	Step ID of the previous step executed

Functions

Function	Description
LowerCase(string text)	Converts input string to lower case.
UpperCase(string text)	Converts input string to upper case.
RegEx(string pattern, string input)	Applies regular expression with input pattern.
StrLength(string text)	Returns the length of input string.
StrTrim(string text)	Returns the input with white-spaces trimmed from both the ends of the string.
StrConcat(...)	Concatenates all the arguments to form a single string and returns the resultant string.
StrReplace(string text, string replace, [any type] with)	Replaces substring in the text with the data and returns the resultant string.
SubStr(string text, int index, int size)	Returns a substring starting from input index upto the defined integer size.
SubStrFirst(string text, int to_index)	Returns substring from index zero to the provided index.
SubStrLast(string text, int from_index)	Returns substring from the provided index to the last index.
Field(string text, string delimeter, int field_num)	Returns a field from the input string separated using the delimeter.
ConvToNumber([any type] data)	Converts the data to number and returns the converted number.
ConvToInteger([any type] data)	Converts the data to integer and returns the integer data.
ConvToFloat([any type] data)	Converts the data to float and returns the float data.
ConvToString([any type] data)	Converts the data to string and returns the string data.
GetFromMap(string key)	Gets value of the input key from the Map Table of the current profile.
SetToMap(string key, string value)	Sets value of the key in the Map Table of the current profile, and also returns the value.
Increment(int number)	Increments the input number by one.
Decrement(int number)	Decrements the input number by one.
Add(int number1, int number2)	Adds the two input numbers.
Sub(int number1, int number2)	Substracts input number2 from input number1.
Mul(int number1, int number2)	Multiplies the two input numbers.
Div(int number1, int number2)	Divides input number1 by input number2.
Mod(int number1, int number2)	Returns the remainder after dividing input number1 by input number2.
Min(...)	Returns the minimum value from the provided arguments.
Max(...)	Returns the maximum value from the provided arguments.
SaveToDB(string dbname, string key, string value)	Stores the input key along with the input value to the specified database and returns the value.
ReadFromDB(string dbname, string key)	Retrieves value of the input key from the specified database.

OK

Alarm Notification Gateway - Help Window

©HipLink Software 2013. All Rights Reserved.

XMPP Gateway

Introduction

XMPP – eXtensible Messaging & Presence Protocol – is an open-standard communications protocol for message-oriented middleware. It is an XML based protocol, in which all the communication between the server and client is done on an XML stream. XMPP follows a client server model where servers can connect to each other to form an XMPP direct connected network. The client first logs into the server, and then they can send message to other clients connected to the same server or to any other server reachable to the client's server. The client can send message in any form with or without control data that can be processed at the other client.

The benefits of using XMPP in HipLink are:

- We can send message and/or control data of any size.
- It provides full security services with TLS and SASL that assures message confidentiality, integrity and authenticity, along with security against various attacks.
- It is an open-standard protocol, which means there are numerous hardware and software vendors available with several products that communicate using XMPP, which can now communicate directly to HipLink and vice versa
- XMPP clients are easily approachable through other XMPP networks, provided that these other XMPP networks are approachable to HipLink
- Most of the famous social IM networks provides XMPP gateway through which HipLink can send message to these social network users. (e.g. Yahoo, MSN, AOL, Facebook messenger, etc.)
- It provides the capability to transfer files between two XMPP clients using p2p protocol, hence it is faster and more reliable. HipLink can use this function to send attachments to HNP/XMPP clients with the message.

XMPP Gateway is an interface to create XMPP user accounts. It gives the user control to enable or disable accounts from the UI. It shows which XMPP accounts and HNP receivers are online along with their presence status.

XMPP Gateway is an optional feature that is available based on the HipLink license key. The panel can be accessed from Settings tab's left navigation bar in Systems menu, as well as from the main Settings panel.

There are three tabs on XMPP Gateway panel:

- Configuration
- Users Account
- Session

©HipLink Software 2013. All Rights Reserved.

Configuration

XMPP Gateway - Configuration Tab

This tab allows the user to configure settings for XMPP Gateway. The tab is further divided into three sections:

- General Settings
- Template Settings
- Advanced Settings

General Settings:

This section is used to setup XMPP Gateway server settings. It contains the following fields:

- **Edit:** Click on this button to display the panel in edit mode.
- **Enable:** Select this checkbox to enable the XMPP Gateway service.
- **Host:** Provide IP address/ server name of the HipLink server where XMPP Gateway is configured. This could be the same HipLink server or a different one who's XMPP Gateway is being used.
- **Service Name:** Provide the name of the service used by XMPP Gateway.
- **Service Port:** Provide the port number used by the service.
- **Server Certificate (for TLS):** Provide the path to the Server Certificate to be used for TLS connection. For the server certificate that comes bundled-in with HipLink, the path is provided by default.
- **Server Private Key:** Provide the path to the Server Private Key used for the above certificate. Leave the default value if you are using the default certificate.
- **Enable Client Contact List:** When this checkbox is selected, the XMPP users of HipLink logged in on an XMPP client will be able to view their contact list. The contact list is the list of all the HipLink receivers whose status is Available at that time, and to whom XMPP users can send messages.

©HipLink Software 2013. All Rights Reserved.

Selecting this checkbox only enables the functionality. To explicitly enable the contact list for an XMPP user, the same checkbox needs to be selected while adding an XMPP account.

Template Settings:

This section is used to setup XMPP message template settings. It contains the following fields:

- **Enable Greetings Message:** Select this checkbox to include greetings message in the message sent by XMPP users to the receivers.
- **Greetings Message:** Define greetings message in this text field.
- **Enable Response Suffix:** Select this checkbox to include suffix in the response action executed by XMPP users on XMPP client.
- **Response Suffix:** Define response suffix in this text field.

Advanced Settings:

This section is used to setup response action execution settings. It contains the following fields:

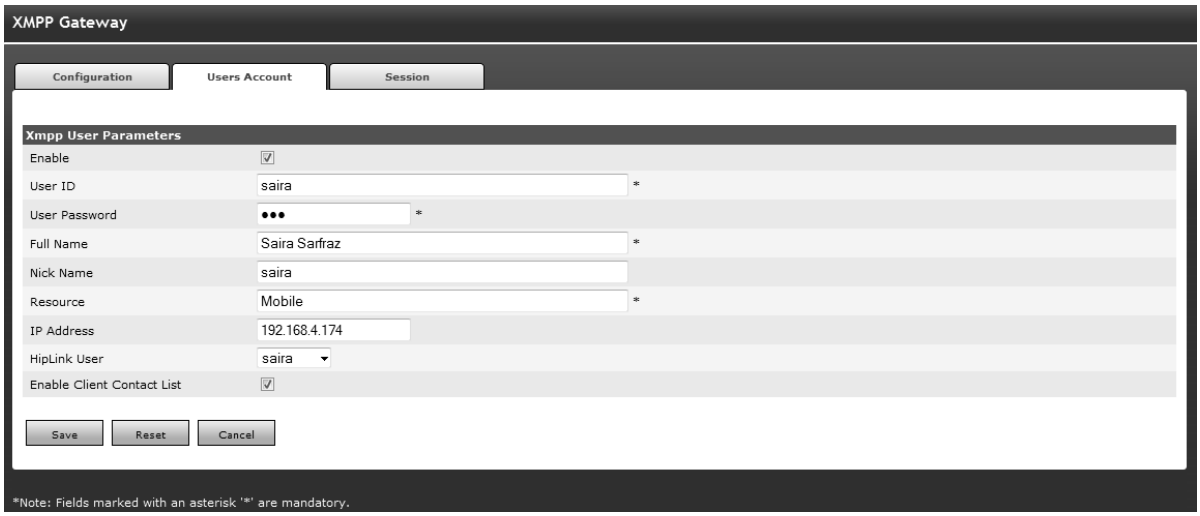
Response Action Execution Thread Pool Size (5-150): Provide the pool size for response action execution thread.

Users Account

XMPP Gateway - Users Account Tab

This tab gives the user control over XMPP user accounts. It allows the user to add, edit, delete, enable, or disable XMPP accounts. This panel is similar to Users panel of HipLink, except that it displays only XMPP accounts and not the native HipLink users. To add an XMPP user account, click on Add User button. This will open a form XMPP User Parameters in the same tab.

©HipLink Software 2013. All Rights Reserved.



XMPP User Parameters

Enable: This checkbox is selected by default. If you want to disable the XMPP account, then uncheck this checkbox.

User ID: User id for the XMPP user account (mandatory).

User Password: Password for the above user id. These are the credentials for an XMPP user to log into an XMPP client (mandatory).

Full Name: Full name of the XMPP user to appear to other XMPP users in communication.

Nick Name: Name to appear to other XMPP users in place of Full Name (optional).

Resource: This information is used during file transfer. An XMPP user can have multiple sessions with a single account. While logging into an XMPP client, user needs to provide resource along with login credentials. When a text message is sent to an XMPP user, it will be received by all the sessions; however, when a message contains a file attachment as well, it would be sent only to the session that has the same resource as defined in this form.

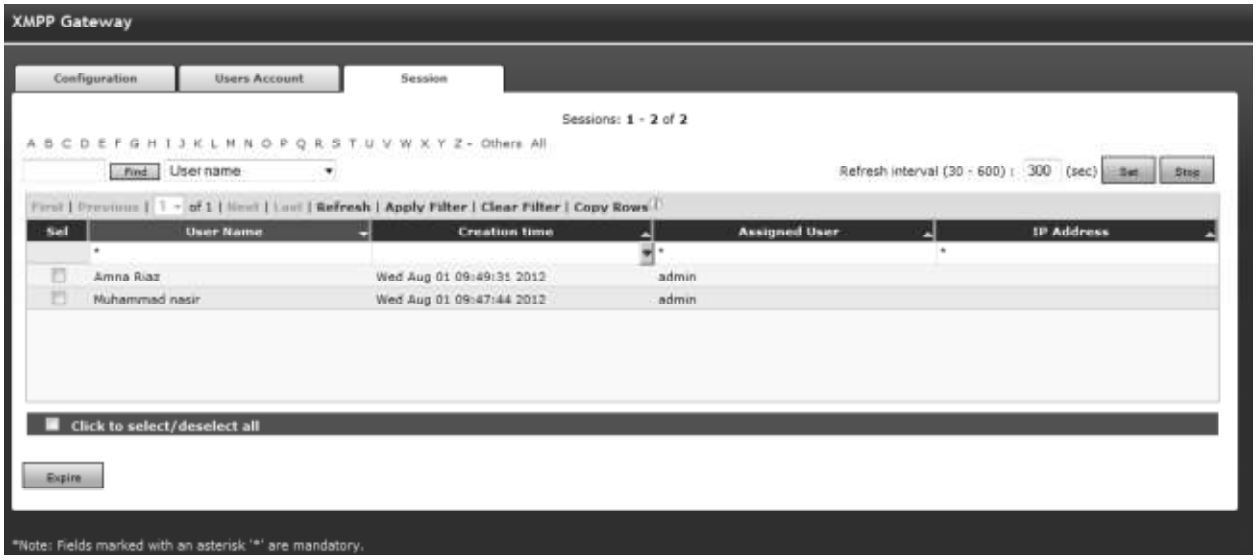
IP Address: IP address from which the XMPP user can log on. This field is optional, but when an IP address is provided in this field, the XMPP user would be allowed to login to an XMPP client only on this IP address.

HipLink User: Select the HipLink user from the dropdown list. The XMPP user will have the same set of permissions – for sending messages to HipLink receivers – as the selected HipLink user.

Enable Client Contact List: Select this checkbox to enable contact list for this XMPP user. This check box is editable only if it is selected on main Users Account tab.

©HipLink Software 2013. All Rights Reserved.

Session



XMPP Gateway - Session Tab

This tab shows the active sessions of XMPP users, as well as allows the user to expire an active session. It contains the following columns:

- **User Name:** User name of the XMPP account.
- **Creation Time:** Time when the session is created on the XMPP client (login time).
  Assigned User: HipLink user assigned to this XMPP account.
- **IP Address:** IP address from which the XMPP user is logged in. This field contains a value only if an IP address was defined for the XMPP user.

To Expire a User Session:

- On Session tab, select the user(s) whose session you want to expire.
- Click on Expire button.
- Click the OK button to confirm expiry, or click Cancel to revoke this action.

The tab refreshes automatically every 300 seconds (by default). User can define a different refresh interval, between 30 to 600 seconds.

The filters and paging defined on this tab are same as on all other panels of HipLink.

©HipLink Software 2013. All Rights Reserved.

Siemens Fire Safety Integration

Introduction

The integration component in the zipped folder acts as a bridge to communicate between the Siemens Fire System and HipLink. The central panel of the fire system is connected to the computer through a serial port (COM port). As soon as the alarm comes on the Fire System, this component relays the alarm to HipLink, allowing the fire emergency notification to be sent to different users through HipLink.

The key idea of how this component works is that the events (Alarm In, Trouble In, etc.) on the Siemens Fire System are assigned to HipLink receiver groups. When the fire event comes from the Siemens Fire System, it has the device or sensor code number along with. The integration component fetches the corresponding name/description assigned to that device. If the system does not reply with the name within a specified time, the message is also sent to HipLink to be delivered to different users. If the system does not give us the name for the device we send the message anyway

In addition to the events from different devices on the Fire System, the integration component keeps polling the Fire System to ensure the connection with the system. The component waits for an acknowledgement from the system for a specified amount of time and if it does not receive a reply within that time frame, a message is delivered to a group assigned for connection problems.

Installing the Component

This chapter contains a step-by-step description on the installation of the component. The component will be installed in the target system with a standard Windows Installer that is accompanied.

1) Install .Net framework 2.0. Run dotnetfx.exe with all default settings.

2) Install SQL Server 2005 Express Edition SP2. Run SQLEXPR32.EXE with all default settings.

3) Install SQL Server Management Studio Express. Run SQLServer2005_SSMSEE.msi with all default settings.

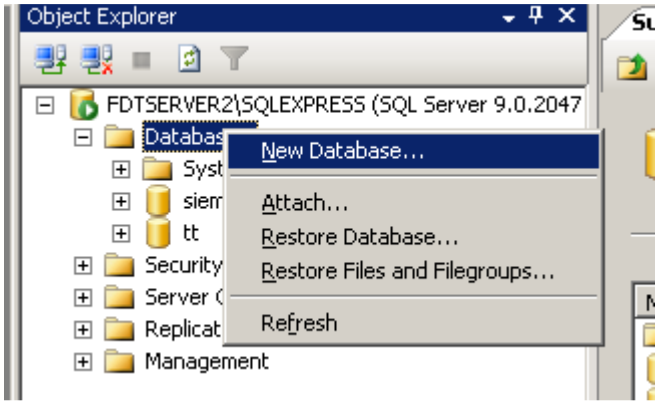
4) Install Enhanced FSI listener installer. Run eFSI_Service_Installer.msi with all default settings.

Creating Database storage for FSI Device Points Custom Information

1) Start SQL Server Management Studio Express
2) On Connect to Server Screen click connect

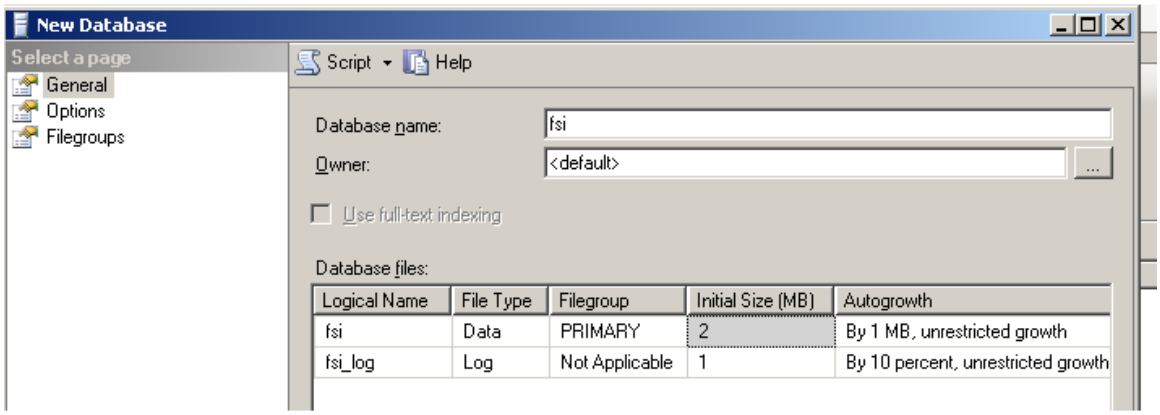
©HipLink Software 2013. All Rights Reserved.

3) Select Databases and right click select New Database.

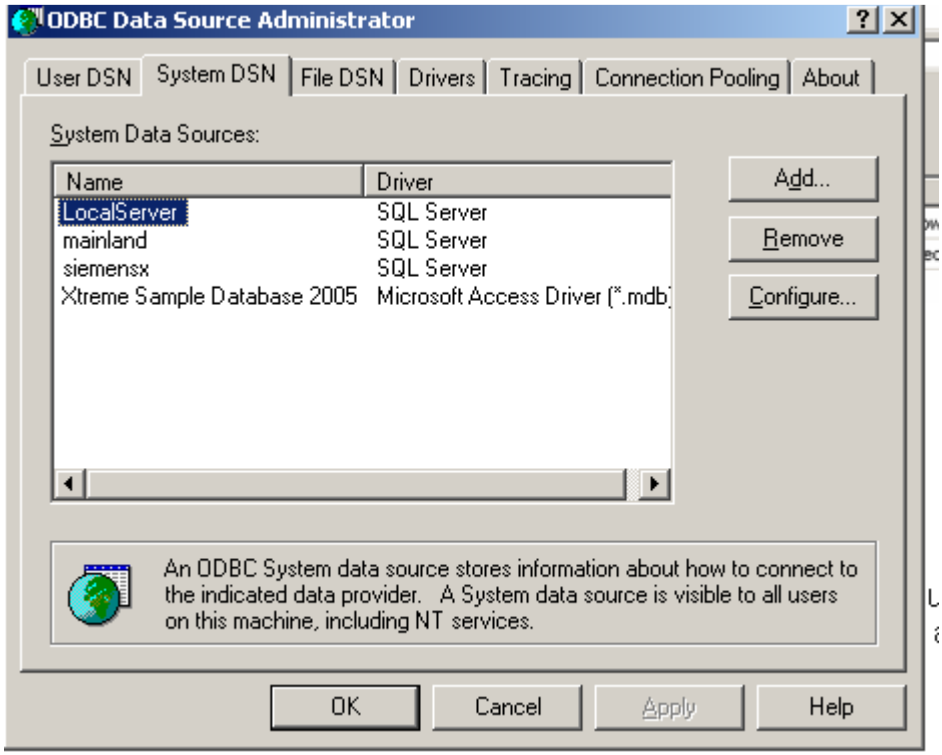


©HipLink Software 2013. All Rights Reserved.

4) On New Database screen type Database Name: fsi and click OK

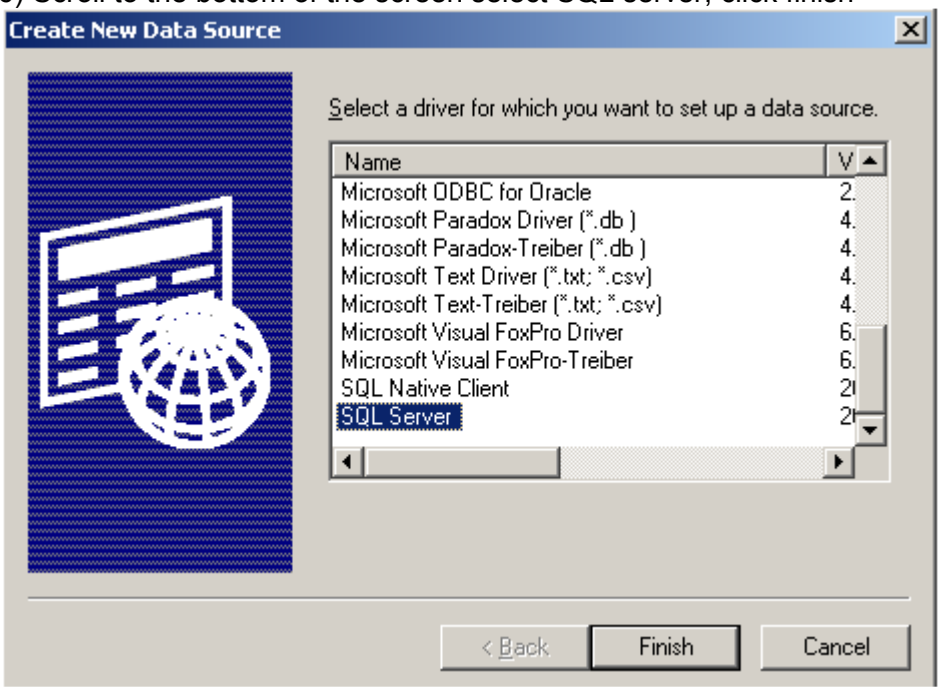


5) Go to Start Control Panel, Administrative Tools, Data Sources (ODBC), System DSN and click Add

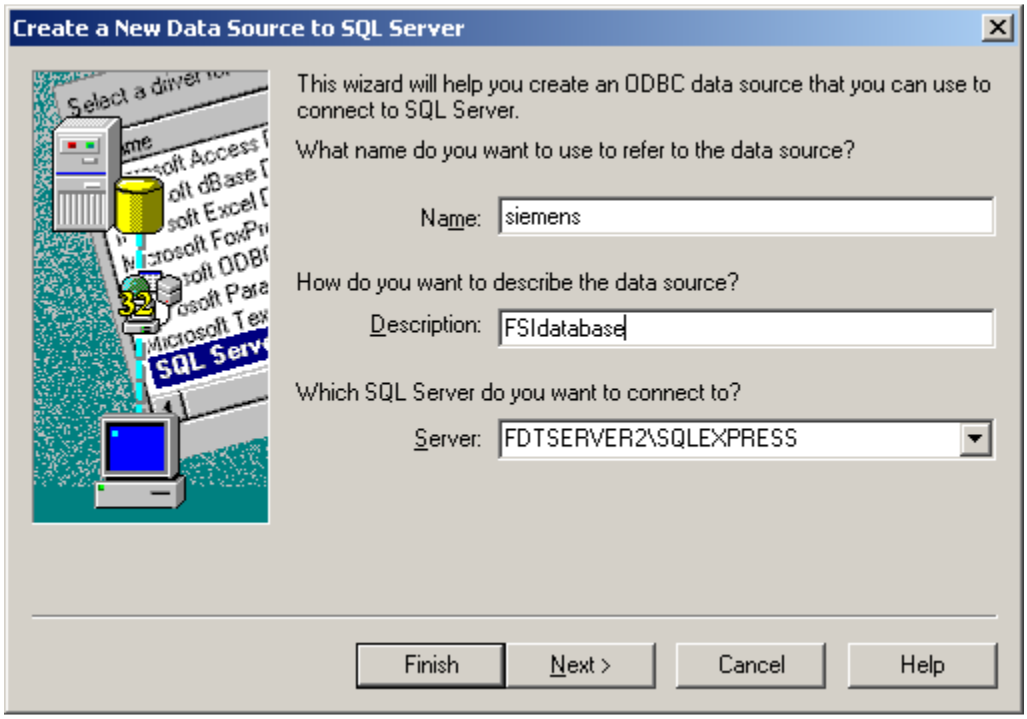


©HipLink Software 2013. All Rights Reserved.

6) Scroll to the bottom of the screen select SQL server, click finish

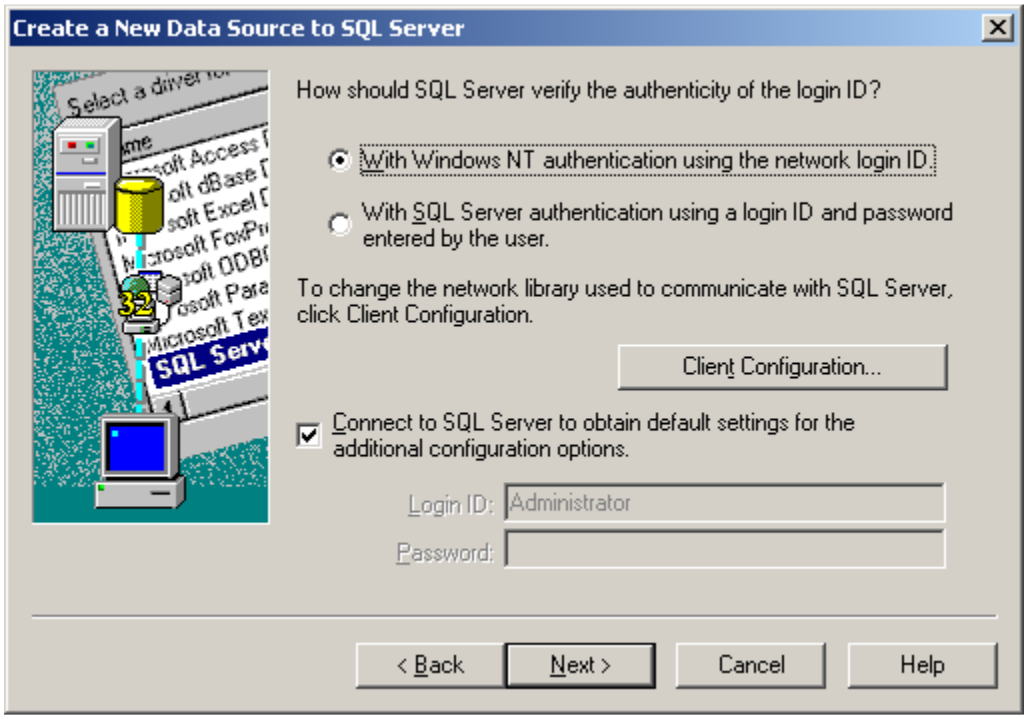


7) type
Name: siemens
Description: optional
Server Name: type same name as shows in server name field on step 2)

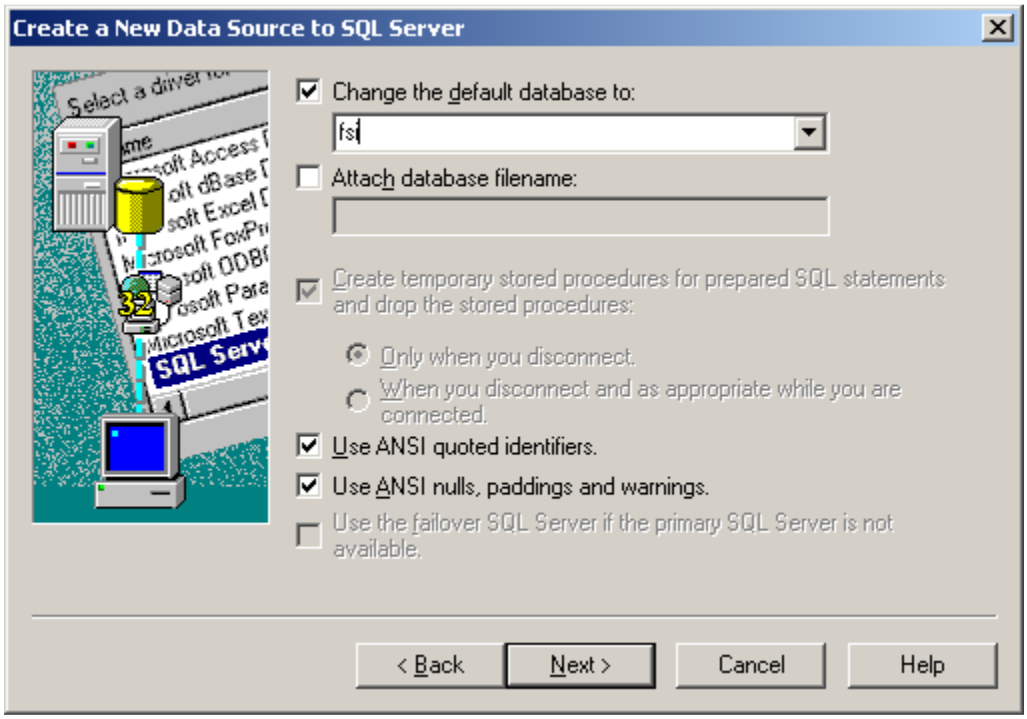


©HipLink Software 2013. All Rights Reserved.

8) Click on next, following screen shows up, click on Next

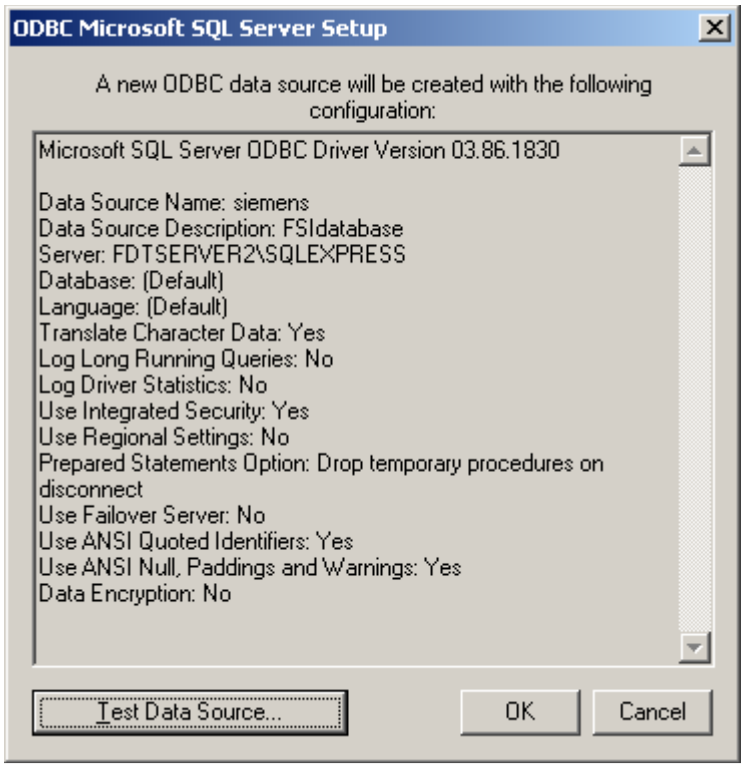


9) On the screen below, check Change the default database to and select fsi; click on Next.



10) Click Finish, the following screen shows up

©HipLink Software 2013. All Rights Reserved.



9) Click on Test Data Source…button. Result should be

10) Go to installation folder (by default C:\Program Files\HipLink\Enhanced FSI Listener Service), right click on siemens.bat file, select Edit Setup parameters as follows and save file.

©HipLink Software 2013. All Rights Reserved.

cscript //NoLogo "siemens.vbs" /dbtype=mssql /DSN=siemens /UID= /PWD=

11) Double click on siemens.bat to run it.

Configuring the component

After the component is installed successfully, it has to be configured manually before it can be used to communicate between the Siemens Fire System and HipLink. This configuration is described in this chapter.

To configure the component:

1. Navigate to the installation folder.
2. Locate the file named *eFSI_Service.ini.*
3. Open the file in a text editor (e.g. Notepad).
4. Make the required changes as described in the following section.
5. Save the file and close it.

The eFSI_Service.INI file:

The parameters in the INI file are grouped in three sections labeled [Default] and [Groups] and [EventFilter]

The [Default] section contains parameters related to installation information of HipLink and the Siemens Fire System. A description of each of the parameters is given below. A sample **eFSI_Service.INI** file is given at the end of this chapter.

Parameter	Description
URL	This is the HipLink URL. i.e. *http://demo.hiplink.com/cgi-bin/action.exe*
Panel	Address of the panel
PanelType	Panel type MXL/XLS
isXNET	Type of system stand alone/networked
ComPort	The serial port (COM port) through which the Siemens Fire System is connected to the computer
BaudRate	A number (0–9) describing the baud rate of the serial port – the speed at which data is transmitted through the serial port. The mapping of the number to the baud rate is as follows. • 0 – 110

©HipLink Software 2013. All Rights Reserved.

Parameter	Description
	• 1 – 300 • 2 – 1200 • 3 – 2400 • 4– 4800 • 5 – 9600 • 6 – 19200 • 7 – 38400 • 8 – 57600 • 9 – 115200
DSN= User= Pwd=	DSN name Leave empty for Windows Authorization Leave empty for Windows Authorization
PollingRate	Interval in milliseconds, between listener module polls of FSI system.
MaxTimeNoAnswer	The maximum amount of time in seconds without an answer from the FSI before an alert is sent to the group assigned to be notified on Bad Connection to FSI.
MaxTimeNoResponseOnSystemData	The maximum amount of time in seconds FSI listener waits for response on System Data Queries.
NodeMapSynchCycle	Time in seconds between Listener module requests to FSI to retrieve system node information
LogLevel	This setting is used to define log level of FSI listener

The [Groups] section contains the names of *valid* HipLink Receiver Groups that will receive notifications. The groups are assigned as explained below.

Parameter	Description
<*GroupName*>	Code numbers associated with different type of messages on the Fire System. Two or more codes are separated by spaces

For example, if “AlarmGroup”, “TroubleGroup”, “SupvGroup” and “ConnectionGroup” are valid HipLink receiver groups, and the Siemens Fire System has messages with code numbers 21, 24, 27, 28, 7f, the [Groups] section may validly contain the following lines.

```
AlarmGroup=21
TroubleGroup=24
SupvGroup=27 28
ConnectionGroup=100
```

Similarly, if “FSI_Receiver_Group” is a valid HipLink receiver group and it has all the message codes 21, 24, 27,28 and 7f, the [Groups] section will contain the following line.

```
FSI_Receiver_Group=21 24 27 28 7f
```

©HipLink Software 2013. All Rights Reserved.

In case of two or more receiver groups assigned the same code, the later one will be considered assigned.

List of supported FSI Messages and assigned codes

eAlarmIn 21
eAlarmOut 22
eAlarmAck 23
eTroubleIn 24
eTroubleOut 25
eTroubleAck 26

eSupvIn 27
eSupvOut 28
eSupvAck 29
eSecurityIn 2a
eSecurityOut 2b
eSecurityAck 2d
eStatusIn 2e
eStatusOut 2f
eAudSil 30
eAudUnsil 31
eSystemReset 32
eTestEventIn 61
eTestEventOut 62
eSystemFlagStateChange 68
eBadConnection 7f

Group [EventFilter] contains settings (yes/no) for all supported FSI events.
Set “yes”, if notification is required on particular event, otherwise “no”
For example:
[EventFilter]
eAlarmIn = yes
eAlarmOut = yes
eAlarmAck = no

Sample FSIListenerService.INI file

[Default]

;hiplink url
URL=http://demo.hiplinkwireless.com/cgi-bin/action.exe

;FSI settings
Panel = 1
PanelType = mxl
isXNET = false
ComPort=1

©HipLink Software 2013. All Rights Reserved.

```
BaudRate=5; follows a table with the valid baud rate values
;110    0
;300    1
;1200   2
;2400   3
;4800   4
;9600   5
;19200  6
;38400  7
;57600  8
;115200 9

;;database settings
DSN=siemens
User=
Pwd=


;;FSI message polling interval in mlseconds
PollingRate = 200; mlsecs

;;the maximum amount of time without an answer from FSI before
;;an alert is sent to Hiplink on Bad Connection in seconds
MaxTimeNoAnswer=10;seconds;

MaxTimeNoResponseOnSystemData = 5;seconds

;;FSI node map and custom message polling interval in seconds
NodeMapSynchCycle = 3600;seconds


;LogLevel = Info
;LogLevel = Error
LogLevel = DeveloperInfo

ConnectionType = Real

[Groups]
AlarmGroup=21 22 23
TroubleGroup=24 25 26
SupvGroup=27 28 29
ConnectionGroup=2a 2b 2d 2e 2f 30 31 32 61 62 68 7f

[EventFilter]
eAlarmIn = yes
eAlarmOut = yes
eAlarmAck = yes
eTroubleIn = yes
eTroubleOut = yes
```

©HipLink Software 2013. All Rights Reserved.

eTroubleAck = yes
eSupvIn = yes
eSupvOut = yes
eSupvAck = yes
eSecurityIn = yes
eSecurityOut = yes
eSecurityAck = yes
eStatusIn = yes
eStatusOut = yes
eAudSil = yes
eAudUnsil = yes
eSystemReset = yes
eTestEventIn = yes
eTestEventOut = yes
eSystemFlagStateChange = yes
eBadConnection = yes

©HipLink Software 2013. All Rights Reserved.