

IM Gateway Plugin

The IM gateway plugin provides the functionality you may be used to from other jabber transport implementations, but uses the internal apis of Openfire to make interaction as smooth as possible. It implements a number of protocols from the same plugin in what it hopefully a seamless fashion. Here you will find design docs, planning docs, and misc info about the plugin and its underlying transports.

?

End User Documentation

End users should either get their XMPP/Jabber server administrator to set up their IM accounts, or they can register with the transports provided by the IM Gateway plugin via normal mechanisms. Typically, normal mechanisms involve opening up your client's service discovery browser window, finding the transport you want to make use of (for example, if you want to log into your AIM account, look for the AIM Transport), right clicking on it, and choosing register. This is, however, implemented in a myriad of ways depending on what client you are using. The instructions I just mentioned are how you do it in Psi. In Spark (as of 2.0) there should be buttons that show up at the top of your contact list that will allow you to register with the transports. Pandion has a separate Transport Browser (might be called Gateway, can't recall). Once you have registered, the transport should detect whether you are online and log you in automatically, at which point you should see your contacts show up in your roster from the legacy network. (for example, in AIM, your AIM buddies should show up) You can communicate with them just like any other contact in your roster/contact list. You will also note that another contact will appear in your roster with the name of the transport. This is used to indicate your status with the legacy network and to provide a mechanism for the transport to "hear" you log in and change status. Please be aware that this may change in the future as we are evaluating "cleaner" ways of doing this.

?

Troubleshooting

Pandion: MSN Contacts showing up as 'No status message'

Unfortunately, this is caused by a weird hack Pandion has in place to deal with an older vintage of some MSN transport. This can be seen in the Pandion directory in src\main\ClientRoster.js. Look for MSN tweak. If you go into settings\default.xml, search for msnworkaround, and set it to false, you will fix the problem.

?

Pandion: Can not add MSN contacts

Unfortunately, this is due to Pandion not making use of the real [Jabber IQ Gateway Protocol](#) as described in [XEP-0100](#). Instead, it converts between MSN IDs and XMPP addresses on it's own in an incompatible "legacy" format. The conversion done via the aforementioned protocol follows the [XEP-0106: JID escaping](#) standard.

?

Server Administrator Documentation

Once you plop the gateway plugin jar in the proper location (plugins directory) on your Openfire server, it should auto-extract (assuming the server is running) just like any other plugin. Once it's done that and has activated itself, you will see a new section appear in the admin console towards the bottom. The Settings section allows you to choose which transports you want to have enabled on your server, and in the future will provide you with the ability to customize certain aspects of it, like restricted access lists, special config options, that sort of thing. Enable the ones you want by clicking the appropriate checkboxes. The Registrations section allows you to see who all is registered with your server, what services the are registered with, what their usernames are on those services, and also lets you add, edit, and delete registrations. Depending on your preference, you may either prefer to set up the registrations for all of your users instead of having them do it themselves. Of course, that typically would require them to give you their password and I never condone that. 😊 For now, that's about all there is to administering the plugin.

?

If you have a firewall set up, you do not need to open any inbound port connections, but you will need to make sure that you can make outgoing connections to the following ports and hostnames:

- ??AIM: login.oscar.aol.com 5190
- ??Gadu-Gadu: appmsg.gadu-gadu.pl 80
- ??GTalk: talk.google.com 5222
- ??ICQ: login.oscar.aol.com 5190
- ??IRC: irc.freenode.net 7000
- ??MSN: messenger.hotmail.com 1863 (also 443 for initial connection)
- ??SIMPLE: localhost 5060 (of course there's a good chance you'll point it elsewhere)
- ??XMPP: jabber.org 5222

??Yahoo: scs.msg.yahoo.com 5050

?

Please be aware that these are only the initial connections made.? Many of the services will connect to other servers for difference aspects of your legacy IM session.? All of these connections should stay on the same port though.? There may be ranges of IP addresses or something that you can open up but I do not know those lists.? Also, it is now possible to change the initial connect host and port via Openfire Properties.

?

There are also some other Openfire system properties that apply to the plugin, which you can see in the appropriate table at [No Transports In Roster](#).

?

Developer Documentation

The source for the plugin is available via [IM Gateway's SVN repository](#). Note that the plugin should be built amongst Openfire's at src/plugins/gateway.? Typically this is done via **ant plugins** (builds all plugins) or if you want to build the gateway plugin specifically, **ant -Dplugin=gateway plugin**. Typically, if you check out Openfire's trunk, you will get the most recent release of the IM Gateway plugin. You can then cd to src/plugin and use svn switch to switch to the plugin's trunk. (you can also svn edit . and edit the properties referring to the gateway plugin's external repository, and then svn update to pull in the changes)

?

The toolkits we are currently using with the IM Gateway plugin are as follows:

??Protocol Support

??? AIM/ICQ (OSCAR) Support: [The Joust Project](#) (specifically joscar) [javadocs](#)

??? Gadu-Gadu Support: [JGGApi javadocs](#)

??? IRC Support: [Martyr javadocs](#)

??? MSN Support: [Java-JML javadocs](#)

??? SIP Support: [Jain-SIP javadocs](#)

??? XMPP/GTalk Support: [Smack javadocs](#)

??? Yahoo Support: OpenYMSG (based off of [jYMSG javadocs](#), hosted amongst Jive's repositories)

?

Helper Libraries

- Concurrent Programming: [concurrent](#) (used by Jain-SIP)
- Direct Web Remoting: [DWR](#) (AJAX support for admin interface)
- I/O Framework: [Cindy](#) (used by Java-JML)
- Lightweight Container: [PicoContainer](#)
- Logging Services: [Log4j](#)
- Regular Expressions: [Jakarta Regexp](#)
- Socks Support: [Java SOCKS Proxy Server](#) (used by joscar)
- XML-RPC Support: [Redstone XML-RPC javadocs](#)

?

XML-RPC Interface

As of version 1.1.0, an XML-RPC interface exists for system administrators or web developers to create custom web based (or whatever) front ends for their users to use to register. Other tasks can be performed via the XML-RPC interface as well. The URL to access the XML-RPC interface is <http://yourserver:9090/plugins/gateway/xml-rpc> (or the recommended https, and adjust for whatever ports you may have set on your web admin interface). Described here is a list of available commands, their arguments, and what they do.

Manager.toggleTransport - Enable or disable a transport

- password** - the password you set `plugin.gateway.xmlrpc.password` to
- transportName** - the type of transport you are toggling (aim, icq, msn, yahoo, irc, xmpp, gtalk, or sip)

Manager.addRegistration - Add a registration to a particular transport

- password** - the password you set `plugin.gateway.xmlrpc.password` to
- user** - username or JID of the XMPP account to register with the transport
- transportName** - the type of transport you are adding to (aim, icq, msn, yahoo, irc, xmpp, gtalk, or sip)
- legacyUsername** - username/account name from the legacy system on the given transport
- legacyPassword** - password used to log into the legacy system on the given transport
- legacyNickname** - nickname used with the legacy system, only used with IRC

Manager.deleteRegistration - Delete a registration from a particular transport

- password** - the password you set `plugin.gateway.xmlrpc.password` to
- user** - username or JID of the XMPP account to be unregistered from the transport

??? **transportName** - the type of transport you are deleting from (aim, icq, msn, yahoo, irc, xmpp, gtalk, or sip)

??? **Manager.updateRegistration** - Updates a registration with a particular transport

??? **password** - the password you set plugin.gateway.xmlrpc.password to

??? **user** - username or JID of the XMPP account to register with the transport

??? **transportName** - the type of transport you are updating on (aim, icq, msn, yahoo, irc, xmpp, gtalk, or sip)

??? **legacyUsername** - username/account name from the legacy system on the given transport

??? **legacyPassword** - password used to log into the legacy system on the given transport

??? **legacyNickname** - nickname used with the legacy system, only used with IRC

??? **Manager.getActiveTransports** - Retrieves a list of active(enabled) transports

??? **password** - the password you set plugin.gateway.xmlrpc.password to

?

Related Docs

To help explain some of the inner workings of transports, the plugin, and misc other things, we also have some 'articles' on various topics:

??? [How Standard Transports Work](#)

??? [No Transports In Roster](#)

?