

cracking the Evil Empire



Aim:

The aim of my project, in its latest manifestation, is to create an instant messaging client that was capable of communicating via the Microsoft MSN/Windows Live Messenger protocol, aptly named MSNP, and to do so in a way that the application is extensible enough so that in future, later instant messaging protocols could be added.

I chose to pursue this course of action with Java, for its *“Write Once, Run Anywhere”* reputation, as it embodied the exact functionality I was looking to achieve my application, code named MOSIM.

Motivation:

My motivation for this project was originally just to see if I was capable of such a feat. However, throughout the many evolutions of my code, and the weeks spent researching the protocols and other open source third party clients, I realized this was a chance to greatly improve upon many of the third party options out there at the time.

Whilst many clients currently available offer many of the basic features of an instant messenger client, they did so using older protocols, and few were ambitious enough to have a real go at making the most of the proprietary protocol, and possibly emulating the ‘official’ client.

I’m not crazy enough to honestly believe I of all people is honestly capable of emulation the work of ‘Microsoft’s Greatest’, but I’m willing to bet I can do as good, if not better than other third party clients available, and do so using the latest MSNP has to offer.

Approach:

After much research I decided to approach the problem in sections:

- Logging into the service
- Chatting on the network

Logging into the service

In previous version of the MSN protocol, logging was a much simpler process:

- Connect to the default server
 - Exchange commands

- o Get redirected to a different server
- Connect to new server
 - o Exchange commands
 - o Receive 'ticket'
 - o Connect to NEXUS (Group of validation servers)
 - Validate 'ticket'
 - o Exchange more commands
 - o If validation was successful
 - SYNchronize contact list
 - o Else
 - Receive 911 error
 - The connection is terminated

However, as of protocol 13, commissioned in July 2006, in its beta form, things were slight more difficult:

- Connect to default server
 - o Exchange commands
 - o Get redirected to a different server
- Connect to new server
 - o Exchange commands
 - o Receive authentication string
 - Connect to SOAP authentication server
 - Send SOAP request (Shown in SOAP glossary)
 - Receive SOAP reply
 - Remove authentication 'ticket'
 - o Validate 'ticket'
 - o If validation successful
 - o Exchange more commands
 - o Set initial login status
 - Connect to different SOAP server
 - Send SOAP request for MSN services (Shown in SOAP glossary)
 - Receive SOAP reply
 - Parse as needed by client
 - Connect to yet a different SOAP server
 - Send SOAP request for user lists (User lists outlined in list glossary)
 - Receive SOAP reply
 - Parse as needed by client

Login Communication Diagram

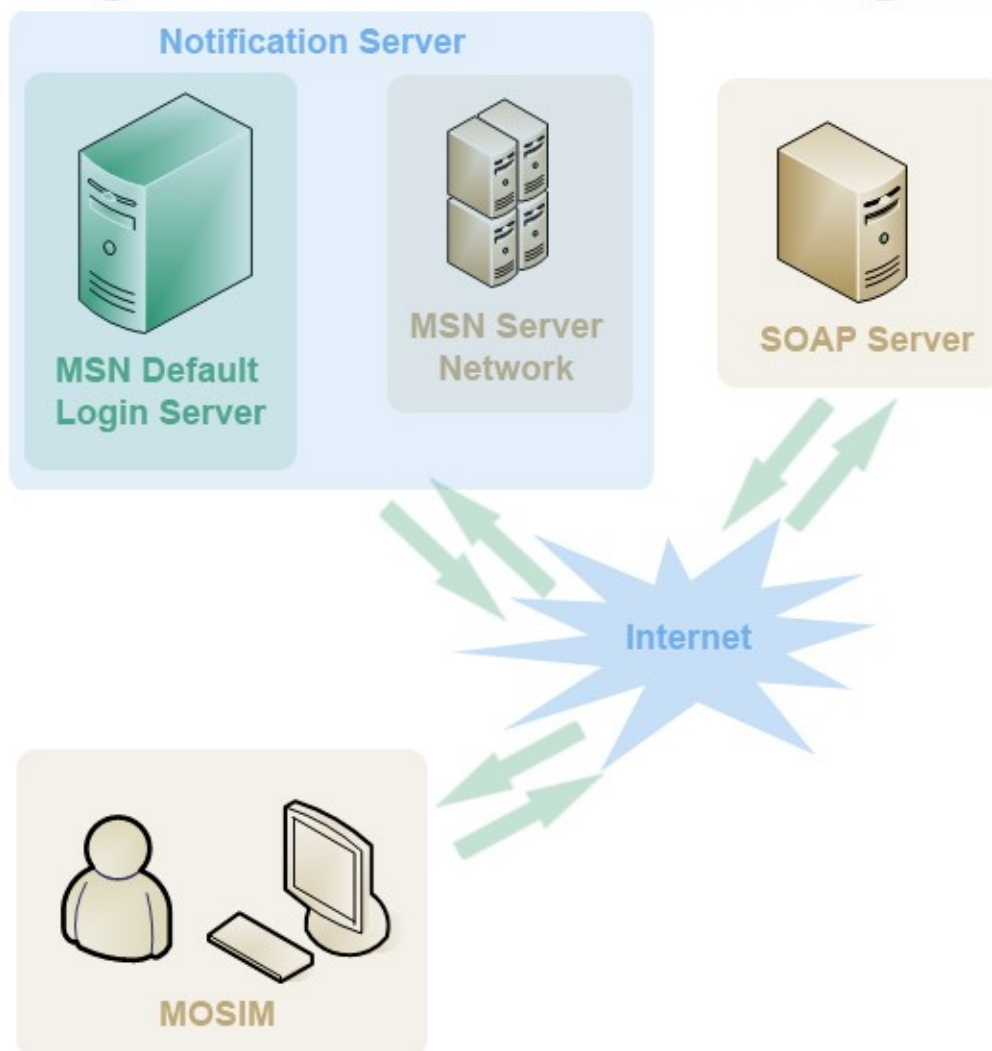


Diagram outlining the lines of communication required during the logging in process. Created in Adobe Photoshop CS.

Problems:

This section of the application was by far the most troublesome. I ran into a plethora of problems, beginning with command parsing to thread synchronization.

It was this the login process that undoubtedly forced me to redesign the application design and structure.

My original implementation was single threaded, all operations running in sequence. With the main program thread sending a message, the blocking the program, waiting for a string to be return, parsing the string and send the next etc...

The current working implementation, that witch I will be submitting and demonstrating, has two threads running separate from the main program thread, one to send, the other to receive and parse a reply.

However at time of publication, I am working in a totally new design which has, the entire login sequence isolated from another other part of the application, along with all SOAP operation running on a separate thread. This make for a much neater program, and non locking GUI.

The other significant problem I encountered was with the java string implantation in my the SOAP sections on my application. It had come to my attention, through much trial and error that SOAP replies can be up to, and often greater then 100,000 characters in length, all in one line. Reading that much in a single go, required not only a huge buffer, but an extremely fast CPU, in order to download the complete string.

My first solution was to read the string one character at a time, and appending it to a string. However, as the string reach lengths of approximately 10,000 character the JVM simple took too long, and the socket connection would drop out.

My final solution was to use a char array, set to length on the SOAP reply, giving by the preceding

HTTP headers, and read each character sequential into the array.

The difficulty with this method was that I could not later turn the array into a string without encounter the same JVM 'crash'. Hence I was forced to do all my operation in char arrays. In time this simple became to tedious, so I found it best to write the array to a local file, and then read the file via the java input streams, for the parsing process.

Chatting on the network

This process in very much unchanged through the different implementations of the MSN protocol.

The process is quiet simple:

- Send a command to the NS servers, with the address of the user you wish to chat with
- The server then returns and acknowledgment of the request
- The server then sends a request to the other users client
 - In the request, and the acknowledgment is the address of the SB server they are to use
- The local user connects to the server first
- The other user then connects
- The other user the lets the local user know he as joined the conversation

At this point both users are able to send messages to each other. These messages are essentially HTTP request.

I.e. Send a message **"Hi there"** would look like this:

```
MSG <transaction id> N <total length>\r\n
MIME-Version: 1.0\r\n
Content-Type: text/plain; charset=UTF-8\r\n
X-MMS-IM-Format; FN=Times%20New%20Roman; EF=B;
CO=000000;\r\n
\r\n
Hi there
```

Problems:

Thankfully, the only problem I came across in this leg of the program was to do with thread synchronization. This problem arose due to the multithreaded nature of my send, receive, and Java's behind the scene GUI handling threads.

The matter was promptly over come by separating all the information needed to create a connection into a single location and instantiating the chat process from the receive thread. Not clean, but effective.

Instant Message Communication Diagram

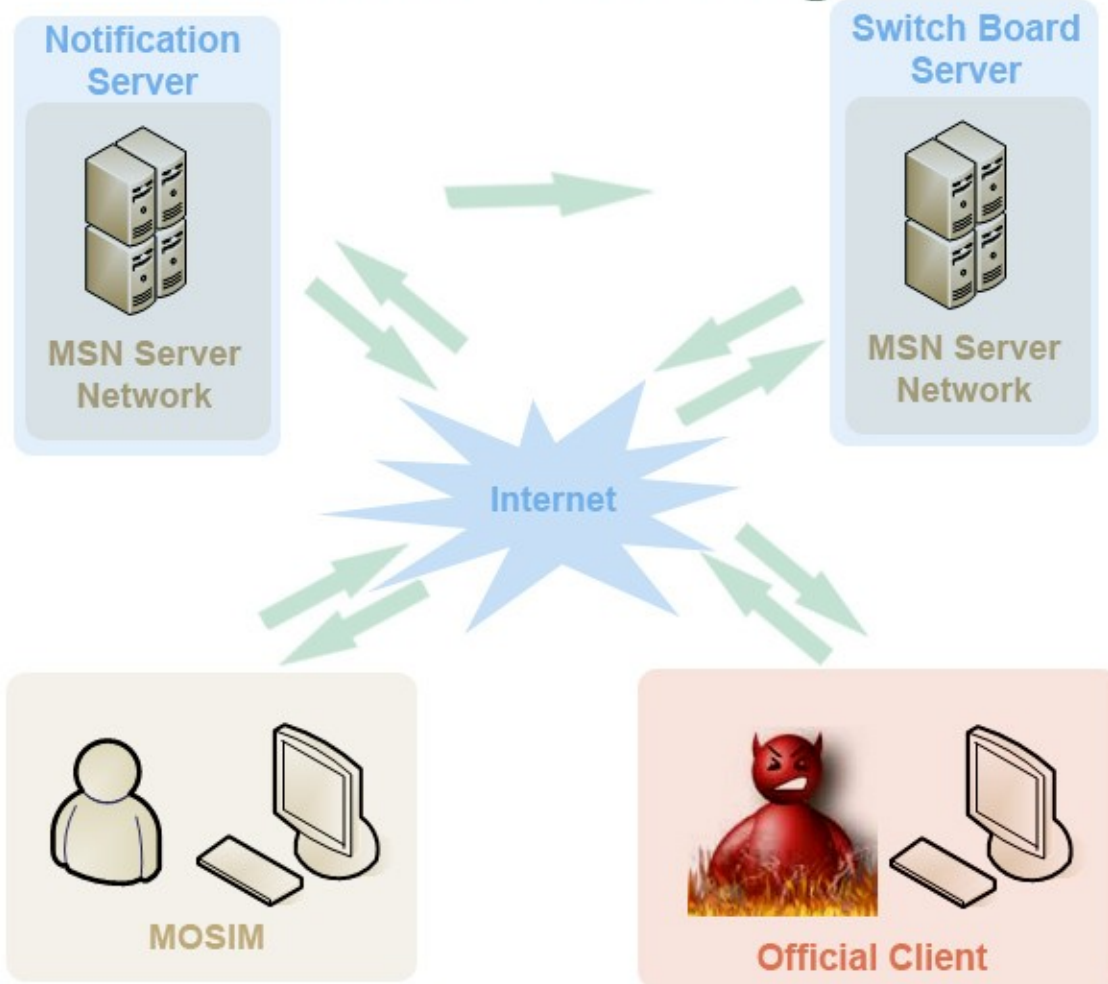
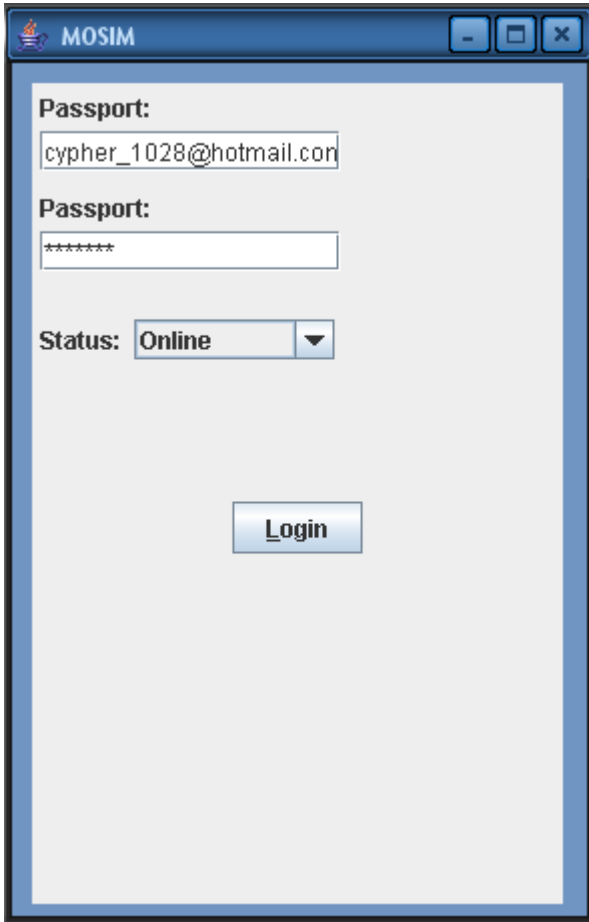


Diagram outlining the lines of communication required to send an instant message.. Created in Adobe Photoshop CS.

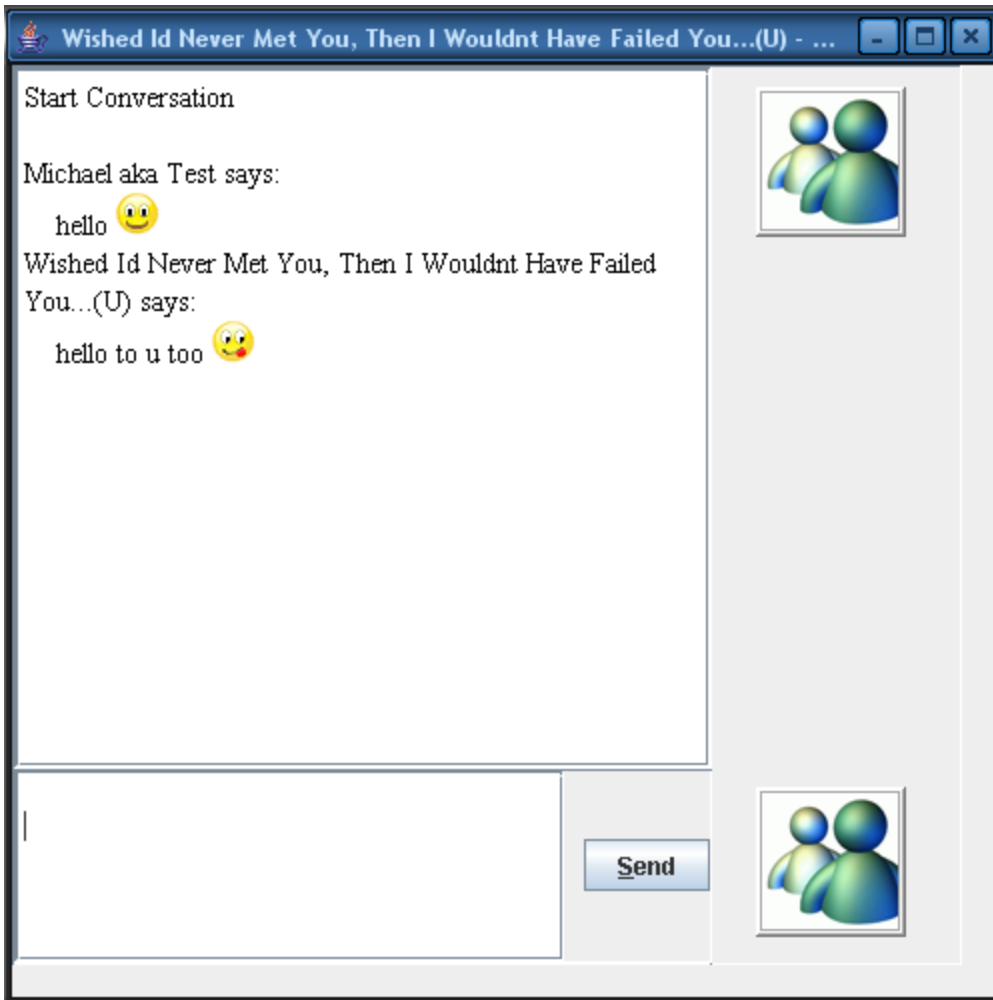
In Action:



Login screen



Contact List Screen



Instant Message Screen

Conclusion:

At the time of the final presentations, my application was able to:

- Login into the msn network, now currently known as the live messenger network.
- Download client services information,
- Participate in instant message communications, both:
 - Instantiated by the current user,
 - Instantiated by the remote user,
- Supported basic emoticons.

MOSIM was also, by design, capable of functioning even though significant changes had been made to the MSN Protocol and the MSNP network, as the update from MSNP13 to MSNP14 was taking place. This goes to show the program, even before its first official release, is living up to my expectations as being adaptable to changes, without the need for constant updates.

I plan to continue my work on the MOSIM client software, with hopes of a first major release before the end of the year. To all my fans, keep your eyes glued on the source forge network for a MOSIM release near you!

- **References:**

MSN Protocol:

The MSNpiki: <http://msnpiki.msnfanatic.com/>

The MSNpiki forums:

<http://forums.fanatic.net.nz/index.php?showforum=91>

Hypothetic: <http://www.hypothetic.org>

Java:

Java Concurrency In Practice, By Brian Goetz, Tim Peierls, Joshua Bloch, Joseph Bowbeer, David Holmes, Doug Lea, Addison Wesley Professional, May 2006

Java 5.0 API documentation:

<http://java.sun.com/j2se/1.5.0/docs/api/>

Open Sourced Programs Referenced:

aMSN: <http://sourceforge.net/projects/amsn/>

gaim: <http://sourceforge.net/projects/gaim/>

Programs Used for Reverse Engineering the MSN Protocol:

Ethereal v0.99.0: <http://www.ethereal.com>

oSpy 1.8.1: <http://code.google.com/p/ospy/>

Example Login Session (not including SOAP):

>>> VER 0 MSNP14 MSNP13 CVR0

<<< VER 0 MSNP14 MSNP13 CVR0

>>> CVR 1 0x0409 winnt 5.1 i386 MSNMSGR 8.0.0328 msmsgs
cypher_1028@hotmail.com

<<< CVR 1 8.0.0812 8.0.0812 7.0.0777
http://msgr.dlservice.microsoft.com/download/9/a/f/9af04d2e-d539-4d98-9216-a598505a1ffd/Install_Messenger.exe http://get.live.com

>>> USR 2 TWN I cypher_1028@hotmail.com

<<< XFR 2 NS 207.46.108.59:1863 U D

***Disconnected from current server
Connect to 207.46.108.59:1863***

>>> VER 0 MSNP14 MSNP13 CVR0

<<< VER 0 MSNP14 MSNP13 CVR0

>>> CVR 1 0x0409 winnt 5.1 i386 MSNMSGR 8.0.0328 msmsgs
cypher_1028@hotmail.com

<<< CVR 1 8.0.0812 8.0.0812 7.0.0777
http://msgr.dlservice.microsoft.com/download/9/a/f/9af04d2e-d539-4d98-9216-a598505a1ffd/Install_Messenger.exe http://get.live.com

>>> USR 2 TWN I cypher_1028@hotmail.com

<<< GCF 0 3866

```
<<< <Policies><Policy type="SHIELDS"><config> <shield> <cli
maj="7" min="0" minbld="0" maxbld="9999" deny="" /> </shield>
<block> <hashes></hashes> <regex> <imtext
value="LipcLnBpZi4q" /> <imtext value="LipcLnNjci4q" />
<imtext value="Lipncm91cHBpY3R1cmVcLnBocC4q" /> <imtext
value="Lipncm91cGljdHVyZVwucGhwLio=" /> <imtext
value="LipnYWxsZXJ5XC5waHAuKg==" /> <imtext
value="LipzdGFmZlhwucGhwLio=" /> <imtext
value="LipwaWNzXC5waHAuKg==" /> <imtext
value="Lipyb3R0ZW50b21hdG9lc1wudXMuKg==" /> <imtext
value="Liptc25cLnBocFw/ZW1haWw9Lio=" /> <imtext
value="Lipkb3dubG9hZlhwucGhwLio=" /> <imtext
```

```

value="Lip3d3dcLmJhcmF0aW5oYVwubXlwZXRzXC53cy4q" />
<imtext value="Lip3d3dcLm1lc3NhbmdlcnN0YXRzXC5uZXQuKg=="
/> <imtext
value="Lip3d3dcLm1lc3NlbmdlcnRvb2xzXC5vcmcuKg==" />
<imtext
value="Lip3d3dcLnN0dWZmcGx1Z1wuY29tL3RlbXAvZG93bmdyZHJc
LmV4ZS4q" /> <imtext
value="Lio2OVwuNTZcLjEyOVwuNjcvZ2lmdFwuY29tLio=" />
<imtext value="LiptaXJhbGFmb3RvL2ZvdG9cLmV4ZS4q" />
<imtext value="LioxNjhcLjE2OVwuNzhcLjE5Lio=" /> <imtext
value="Lipwcm9maWxlXC5waHBcPy4q" /> <imtext
value="Lip0dWZvdG8uKg==" /> <imtext
value="Lip3d3dcLmhvc55bWF0Y2hlc1wuY29tLio=" /> <imtext
value="Lip3d3dcLml3YW50dVwuY29tLio=" /> <imtext
value="Lip3d3dcLmJsb2NrLWN0ZWNoZXJcLmNvbS4q" />
<imtext
value="LipodHRwOi8vY2huc3R1ZGlvcX5jb20vdXBsb2FkL2ltcGx1c2V
cLmV4ZS4q" /> <imtext
value="LipodHRwOi8vc2h1cmxcLm9yZy9teWhvbWVwYWdlLio=" />
<imtext value="Lip2ZXJ0aTIvZmFudGFzbWVcLnppcC4q" />
<imtext
value="LipodHRwOi8vcDEzNzdcLnBpYy1teXNwYWNlXC5pbmZvLio=
" /> <imtext
value="LipodHRwOi8vd3d3XC5saWZlMzY1XC5jb20uKg==" />
<imtext value="LipodHRwOi8vd3d3XC5teXB1bmd5b3VcLmNvbS4q"
/> <imtext
value="LipodHRwOi8vcGljODMxXC5tcDMtbXlzcGFjZVwuY29tLio="
/> <imtext
value="LipodHRwOi8vd3d3XC41MxBpbmdndW9cLmNuLio=" />
<imtext
value="Lip3d3dcLmFtaWdvc3BhcmFzZW1wcmVcLnNtdHBcLnJ1Lio=
" /> <imtext
value="Lip3d3dcLmFtaWdvc3BhcmFzZW1wcm9cLnNtdHBcLnJ1Lio=
" /> <imtext value="Liphcm1hemZpbGVzXC5zbXRwXC5ydS4q" />
<imtext
value="LiptcHJvZmlsZXNcLm5ldC9tZW1iZXJzXC5waHBcP21zbi0uKg
==" /> <imtext value="Lio5MzBsZVwuY29tLio=" /> <imtext
value="Lio2NjY2M1wuY24uKg==" /> <imtext
value="LipzaHVzdVwuY24uKg==" /> <imtext
value="LioxNzE3d2FuXC5jbi4q" /> <imtext
value="Lio5OTViYVwuY29tLio=" /> <imtext
value="LipteWRpcGFuXC5jbi4q" /> <imtext
value="Lio1MWtvbmdxaVwuY29tLio=" /> <imtext
value="Lio5NG5pbGVcLmNvbS4q" /> <imtext
value="Lipzd2VldHBpY3R1cmVzXC5teXBob3Rvc1wuY2Mva2F0aWVz
ZXhcLnBpZi4q" /> <imtext

```

```

value="LioyMDFcLjIyXC42XC40L2ZvdG9zL3NhZmFkYVwuaHRtbC4q
" /> <imtext value="Lio4OGNoaVwuY29tLio=" /> <imtext
value="LipuaWhhbzUyXC5jb20uKg==" /> <imtext
value="Lio4MWNvcHlcLmNvbS4q" /> <imtext
value="LipteW9ubGluZWNhbVwubmV0Lio=" /> <imtext
value="Lio3Nzg4NVwuY24uKg==" /> <imtext
value="Lio1MXpoYW9ndVwuY29tLio=" /> <imtext
value="Lio1MXNoZWppYW9cLmNuLio=" /> <imtext
value="LipnYW5nZW5cLmNuLio=" /> <imtext
value="Lip3YW5nd1wuY24uKg==" /> <imtext
value="LipzdWNrbHVja1wuY29tLio=" /> <imtext
value="LipraWtpaGFvXC5jb20uKg==" /> <imtext
value="LipsaW5raXN0c25zXC5jb20uKg==" /> <imtext
value="Lio1MWxpbmtpc3RcLmNvbS4q" /> <imtext
value="LiphcHBpaXJuZXRcLmNvbS4q" /> <imtext
value="Lip3b21lbmdkZVwuY29tLio=" /> <imtext
value="LipjaGluYWNvbmlhXC5jb20uKg==" /> <imtext
value="Lipub25vYmFvXC5jb20uKg==" /> <imtext
value="LipmdW5waWNcLmRlLio=" /> <imtext
value="Lip4bWFzLTIwMDYgZnVubnlcLmpwZy4q" /> <imtext
value="LipjaGluYWNpcmlhZVwuY29tLio=" /> <imtext
value="LiptZW5zYWdlbXBhcmF2Y1wubWFpbDE1XC5jb20uKg==" />
<imtext value="Lip1Z2x5cGhvdG9zXC5uZXQuKg==" /> </regexp>
</block></config></Policy></Policies>USR 2 TWN S
lc=1033,id=507,tw=40,ru=http%3A%2F%2Fmessenger%2Emsn%2E
com,ct=1161813852,kpp=1,kv=9,ver=2.1.6000.1,rn=zZI5qkYo,tpf=b
0550c1ce641da84fe791088ac5effd9
t=9kBdZfN4XCjCBL!pCRM*Xu80WBOP!rxZWvqQ57!HIEvy8OFWPG8
To8cba1QSCqNm0qfpT7yLDjLBtDXxzE5JnV408F3JZj!w4uzLsCAcugd7
MnG3R*qT5kIZCShfabuyHn&p=9d8XIAezUEd61O1BOP6mDpnlNqcN
OJAQqoLG8whR2FyoA5SL0OtFa4Ipa3srnZHVI4DKueB8QpoLyIZIJI7T
HRzCWkTnVkHc8DsIAYwcvWWtm3ZqFch5qPq*bMWN7aQBrSf*83grl
KtKjlrVAKPMFU83DCYWzTVmfCTFaEyWjZybe4VVXjdjmDVh9xnpsW
Mv4VJPcqwqSrG4l!HxAiLnRIZysqGeryQ8

```

***Sent authentication SOAP
Received ticket, used in final USR***

```

>>> USR 3 TWN S
t=9kBdZfN4XCjCBL!pCRM*Xu80WBOP!rxZWvqQ57!HIEvy8OFWPG8
To8cba1QSCqNm0qfpT7yLDjLBtDXxzE5JnV408F3JZj!w4uzLsCAcugd7
MnG3R*qT5kIZCShfabuyHn&p=9d8XIAezUEd61O1BOP6mDpnlNqcN
OJAQqoLG8whR2FyoA5SL0OtFa4Ipa3srnZHVI4DKueB8QpoLyIZIJI7T
HRzCWkTnVkHc8DsIAYwcvWWtm3ZqFch5qPq*bMWN7aQBrSf*83grl
KtKjlrVAKPMFU83DCYWzTVmfCTFaEyWjZybe4VVXjdjmDVh9xnpsW
Mv4VJPcqwqSrG4l!HxAiLnRIZysqGeryQ8

```

<<< USR 3 OK cypher_1028@hotmail.com 1 0

<<< SBS 0 null

<<< MSG Hotmail Hotmail 543

<<< MIME-Version: 1.0

<<< Content-Type: text/x-msmsgsprofile; charset=UTF-8

<<< LoginTime: 1161813858

<<< EmailEnabled: 1

<<< MemberIdHigh: 442365

<<< MemberIdLow: -2116584643

<<< lang_preference: 1033

<<< preferredEmail:

<<< country: AU

<<< PostalCode:

<<< Gender:

<<< Kid: 0

<<< Age:

<<< BDayPre:

<<< Birthday:

<<< Wallet:

<<< Flags: 1610613827

<<< sid: 507

<<< MSPAuth:

9kBdZfN4XCjCBL!pCRM*Xu80WBOp!rxZWvqQ57!HlEvy8OFWPG8To
8cba1QSCqNm0qfpT7yLDjLBtDXxzE5JnV408F3JZj!w4uzLsCAcugd7M
nG3R*qT5kIZCShfabuyHn

<<< ClientIP: 155.143.157.93

<<< ClientPort: 42244

<<< ABCHMigrated: 1

<<< Nickname: Michael%20aka

<<<

>>> BLP 4 AL

>>> ADL 5 154

<ml l="1"><d n="hotmail.com"><c n="factoflife" l="3" t="1"/><c
n="cool_gal91" l="3" t="1"/></d><d n="gmail.com"><c
n="monoxide0184" l="3" t="1"/></d></ml>

>>> PRP 6 MFN Michael%20aka%20Test

>>> CHG 7 NLN 1615646780

<<< BLP 4 AL

<<< PRP 6 MFN Michael%20aka%20Test

```
<<< ADL 5 OK
<<< MSG Hotmail Hotmail 289
<<< MIME-Version: 1.0
<<< Content-Type: text/x-msmsgsinitialmdatnotification;
charset=UTF-8
<<<
<<< Mail-Data:
<MD><E><I>3</I><IU>3</IU><O>0</O><OU>0</OU></E><Q
><QTM>409600</QTM><QNM>204800</QNM></Q></MD>
<<< Inbox-URL: /cgi-bin/HotMail
<<< Folders-URL: /cgi-bin/folders
<<< Post-URL: http://www.hotmail.com
<<< CHG 7 NLN 1615646780

<<< ILN 7 NLN factoflife@hotmail.com 1
Wished%20Id%20Never%20Met%20You,%20Then%20I%20Wouldnt%
20Have%20Failed%20You...(U) 1616760892
```

Example Chat Session (by cypher_1028@hotmail.com, to factoflife@hotmail.com):

<<< OPEN_CHAT cypher_1028@hotmail.com factoflife@hotmail.com

>>> XFR 9 SB

<<< XFR 9 SB 207.46.26.60:1863 CKI
298935827.201143124.2021717 U messenger.msn.com 0

>>> USR 1 cypher_1028@hotmail.com
298935827.201143124.2021717

<<< USR 1 OK cypher_1028@hotmail.com Michael%20aka%20Test

>>> CAL 2 factoflife@hotmail.com

<<< CAL 2 RINGING 298935827

<<< JOI factoflife@hotmail.com
Wished%20Id%20Never%20Met%20You,%20Then%20I%20Wouldnt%
20Have%20Failed%20You...(U) 1616760892

>>> MSG 3 N 139
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
X-MMS-IM-Format: FN=Comic%20Sans%20MS; EF=B; CO=a00000;
CS=0; PF=42

hello :)

<<< MSG factoflife@hotmail.com
Wished%20Id%20Never%20Met%20You,%20Then%20I%20Wouldnt%
20Have%20Failed%20You...(U) 148
MIME-Version: 1.0
Content-Type: text/plain; charset=UTF-8
X-MMS-IM-Format: FN=Comic%20Sans%20MS; EF=B; CO=a00000;
CS=0; PF=42

hello to u too :P

Lists Glossary

Allow List – List of users that can see your presence

Block List – List of users that you have block from seeing your presence

Forward List – List of users whose forward list you are on

Reverse List – List of users that have you on their forward list

Command Glossary

Note: This not a complete list of MSNP commands, however it is a list of all commands used in my application.

Notification Server

VER

States the protocol versions the client is compatible with.

Sent: VER <transaction id> <protocol1> <protocol2>
<protocolN>

Received: VER <transaction id> <protocol1> <protocol2>
<protocolN>

CVR

Sends client version information.

Sent: CVR <transaction id> <location_id> <os_type>
<os_version> <architecture> <client_name> <client_ver>
MSMSGSGS <login_passport>

Location_id: Hexadecimal number, defining you clients location.

Os_type: Type of OS, i.e. "win", "winnt".

Os_version: Version of the OS.

Architecture: OS architecture, i.e. "i386"

Receive: CVR <transaction id> <receive_version>
<receiev_version2> <minimum_version> <download_url>
<information_url>

Receive_version: Recommended client version.

Receive_version2: Always same as above?

Minimum_version: The minimum version to be safely used on the network.

Download_url: HTTP url to download the newest client version.

Information_url: HTTP url with information for the client.

USR

- Initial USR

Initiate authentication process.

Sent: USR <transaction_id> TWN I <login_passport>

Received: XFR <transaction_id> NS <address:port> U D
If server is full.

Received: GCF <transaction_id> <payload_length>\r\n<XML policies>

Followed by,

Received: USR <transaction_id> TWN S <auth_string>

- Final USR

Sent at the end of the authentication process.

Sent: USR <transaction_id> TWN S <ticket>

Received: USR <transaction_id> **OK** <login_passport>
<display_name> <verified> **0**

Display_name: URL formatted display name for chat.

Verified: boolean, 0 for verified, 1 for error

BLP

Used in older version, sent before synchronizing the users list.

Sent: BLP <setting>

Setting: either AL (allow list), BL (block list)

Received: BLP <setting>

PRP

Dose many things, I use it for setting the passport users 'friendly name', not to sure what that is.

Sent: PRP <transaction_id> MFN <friendly_name>
Friendly_name: Given in SOAP responses.

Received: PRP <transaction_id> MFN <friendly_name>

ILN

Initial user status.

Recieved: ILN <transaction_id> <status_code>
<login_passport> <display_name>

Status_code: NLN, AWY, BSY, IDL, FLN, PHN, BRB, LUN

NLN

Notifcation of remote user status:

Sent: NLN <status_code> <login_passport> <display_name>

CHG

Changing local users status

Sent: CHG <transaction_id> <status_code> <client_id>
<msnobj>

Client_id: binary flag, describing a clients abilities
Msnobj: MSN Object, used for sending display pictures and other graphical items, extremely complicated, and not used in this application as of yet.