NASDAQ Compressed Data Feed Option

For its compressed feed options, NASDAQ is reduce the network bandwidth requirement by 75% while retaining the fast and reliable service to which ITCH customers are accustom.

NASDAQ has built its compressed data feed option on the SoupTCP protocol standard, with all downstream communications wrapped in a standard zlib deflator stream.   By doing this, direct feed subscribers are able to implement the NASDAQ compression option by adding a zlib inflator in front of the existing SoupTCP 2.0 reader. Upstream communications remain unchanged from the SoupTCP 2.00 spec.

For more detailed examples see the zlib documentation at: http://www.zlib.net/.

Java Example:

```
if( compressed ) {

InputStream i = new BufferedInputStream(new
InflaterInputStream(sock.getInputStream())); } else

{

InputStream i = new BufferedInputStream(sock.getInputStream());

}
// call i.read() just like a normal soup / ITCH socket
```

C Example:

```
#include "zlib.h"

z_stream d_stream;

/* after initializing d_stream appropriately: */
if (inflateInit(&d_stream) != Z_OK)
/* error */

/* read from socket */
socketReadCnt = read( socketHandle, compressedBuf, compressedBufLen);

/* set up d_stream next_* and avail_* appropriately */
if (inflate(&d_stream, Z_NO_FLUSH) != Z_OK)
/* error */
/* do normal Soup processing */
```

Perl Example:

```perl
use Compress::Zlib ;
use IO::Socket::INET;

my ($iStr, $status) = inflateInit();
my $sock = IO::Socket::INET->new( PeerAddr => $ARGV[2], PeerPort =>
$ARGV[3], Proto => 'tcp', Timeout => 3 ) or die "Can't open socket to
itch port";

# insert code to send soup login


# read itch!

while( $sock->read( $readBuf, 4096 ) > 0 ) {
($unBuf, $status) = $iStr->inflate($readBuf);

# $unBuf is normal itch data.

print "$unBuf";
}
```