Voyager Self Check test perl script

- **Product:** Voyager
- **Relevant for Installation Type:** Multi-Tenant Direct, Dedicated-Direct, Local, TotalCare

Question

How to access the Voyager self check test perl script.

Answer

The perl script attached to this Article can be helpful for understanding and troubleshooting the Voyager SIP2 implementation, when used in conjunction with the 3M SIP Documentation (this document is included with the attached zip file).

Note that to use this script, you must have SSH access to your Voyager server and be able to login as the voyager user. Run the script locally on your Voyager server.

The test script communicates with Voyager Self Check using SIP2 over a telnet connection.

To activate on-screen message reports such as failure error messages see: SELFCHK: How to activate on-screen messages displayed in Self-Check

When you run the script, you will be prompted for the following values for your server and Circulation configuration (the following is an example using the word "Circ" for the Circulation variables):

- Server IP: (usually localhost when run locally on the Voyager server)
- Selfchk Port: (Usually 7031)
- Selfchk User: (Circ)
- Selfchk Password: (Circ)
- Selfchk Location: (Circ)

Instructions:

1. Update the "my @stanza" field appropriately to have (for example) a valid patron barcode in the AA field of the 63 message[1,2]
2. Add any additional messages required
3. /m1/shared/bin/perl SelfCheckTest.pl to run
4. NOTE the \r is carriage return

[1]Example of single message:

```
# Change if needed
my @stanza = (
        "6300120071019    121515          AO|AA7654321|AC|AD|BP|\r"
);
```

[2]Example of multiple messages:

```
# Change if needed
my @stanza = (
        "6300120071019    121515          AO|AA7654321|AC|AD|BP|\r",
        "2300120211110 131858AO|AA7654321|AC|AD|\r",
        "11YY20210708    16070820210808    160708AO|AA7654321|AB19755|AC|\r"
);
```

Additional Information

The following is an example of running a single "63" message - example as per instructions provided above (the user is prompted for Circulation variables). The response is the "64" message:

> [VGER] voyager@us-voysuplnx01 : voyager/ $ perl SelfCheckTest.pl
> Server IP: localhost
> Selfchk Port: 8131
> Selfchk User: Circ
> Selfchk Password: Circ
> Selfchk Location: Circ
> BEGIN Login
> SENDING: 9300|CNCirc|COCirc|CPCirc
> RECEIVING: 941
> SENDING: 9900802.00
> RECEIVING: 98YYYYYN99800320210503   0953312.00AO|BXYYYYYNYYYYYNNNYN|
> END Login
> BEGIN Stanza Loop
> SENDING: 6300120071019   121515   AO|AA7654321|AC|AD|BP|
> RECEIVING: 64   00020210503   0953320000000000030005000000000AO|AA7654321|AEJoe Smith|CA0001|CB0001|PB19850302|BLY|BHUSD|BV0.00|CQN|BEjoesmith@gmail.com|BD1234 Mystreet Ct|
> END Stanza Loop

The following is an example of a complete *script file*[3] in its entirety that runs the 63 message and produces the preceding output:

```
#!/bin/perl -w
use strict;
use Socket;
print "Server IP: ";
my $Ip = <STDIN>;
chomp($Ip);
print "Selfchk Port: ";
my $Port = <STDIN>;
```

```
chomp($Port);
print "Selfchk User: ";
my $User = <STDIN>;
chomp($User);
print "Selfchk Password: ";
my $Pass = <STDIN>;
chomp($Pass);
print "Selfchk Location: ";
my $Loc = <STDIN>;
chomp($Loc);
my $Host = inet_aton($Ip);
# Change if needed
my @stanza = (
        "6300120071019    121515          AO|AA7654321|AC|AD|BP|\r"
);
socket(SOCK, PF_INET, SOCK_STREAM, getprotobyname('tcp')) || die $!;
my $sin = sockaddr_in ($Port, $Host);
connect(SOCK, $sin) || die $!;
my @loginlines = (
        "9300|CN$User|CO$Pass|CP$Loc\r" ,
        "9900802.00\r"
);
my $buf;
print "BEGIN Login\n";
foreach (@loginlines) {
        printf "SENDING: %s\n", $_;
        syswrite(SOCK, $_, length($_));
        sysread(SOCK, $buf, 2000);
        printf "RECEIVING: %s\n", $buf;
}
print "END Login\n\n";
print "BEGIN Stanza Loop\n";
foreach (@stanza) {
        printf "SENDING: %s\n", $_;
        syswrite(SOCK, $_, length($_));
        sysread(SOCK, $buf, 2000);
        printf "RECEIVING: %s\n", $buf;
}
print "END Stanza Loop\n\n";
close(SOCK);
exit 0;
```

[3]The above file is also available to download:

SelfCheckTest-Good-Patron-Info.pl

Self check is an extension module for Voyager - it is not part of the core Voyager software. Activating Self check/SIP2 may require the generation of a quote to cover the implementation and maintenance of this extension.

- **Article last edited:** 17-Feb-2020