

CICS Transaction Gateway Ping Utility

Current Version: v1.3, 27/10/08

Authors: Simon Knights, Russell Wilson, Phil Wakelin, IBM UK, Hursley

Description

This SupportPac provides a simple network ping utility that can be used to test network connectivity and response times to a CICS Transaction Gateway and the CICS systems it is configured to use.

Skill level required

CICS or network administrators with understanding of the IP address and CICS systems used by the CICS Transaction Gateway.

Supplied files

- ch50.pdf - Documentation
- ctgping.jar - JAR file containing Java source and compiled byte code
- ctgping.bat – Windows bat file to run ctgping
- ctgping – UNIX shell script to run ctgping
- ctgping.jcl - JCL to run ctgping as a batch job
- pong.ccp – COBOL COMMAREA CICS back-end program.
- ponglong.ccp – Channels and Containers CICS back-end program
- license – Directory for translated license files

Return codes

When running on a distributed platform or under Unix System Services on z/OS the ctgping utility returns the following return codes on completion:

RC_OK = 0;

INVALID_PARAMETERS = 4;

OPEN_FAIL = 8;

RC_REQUEST_FAIL = 12;

CLOSE_FAIL = 16;

Return codes are not available when ctgping is invoked via the sample JCL. Ctgping is invoked by the CTGBATCH EXEC in the PING step in ctgping.jcl. This step does not set a return code if ctgping fails.

New function in v1.3

The following new function is provided in this version:

- Support for channels and containers.
- Output CICS transaction abend codes
- Variable data length on requests

New function in v1.2

The following fixes are provided in this version:

- Return codes propagated back from the ctgping shell script on distributed platforms
- Userid and password can be specified in any order

New function in v1.1

The following new function and fixes are provided in this version:

- Return codes for system automation
- User ID and password options for connection to a secure server
- A configurable delay between each iteration of ping requests
- z/OS sample JCL
- Last parameter corrupted on Linux
- Windows .bat file contained extraneous .class statement.

Installation:

1. To install the SupportPac, copy the file ch50.zip to a temporary directory on your chosen system and uncompress using unzip.
2. Ensure that your CLASSPATH includes a valid version of the CICS Transaction Gateway ctgclient.jar. This JAR file is supplied with the CICS TG on all platforms. The JAR file can be found in the classes subdirectory of a CICS TG install or is supplied in SupportPac CC03.

For example on Windows issue the following command:

```
SET CLASSPATH=  
C:\Program Files\IBM\CICS Transaction Gateway\classes\ctgclient.jar
```

3. Update your PATH variable to include a valid Java environment. Note that this SupportPac is built with IBM SDK 1.4.2 and requires a JVM at V1.4 or later. For example on Windows issues the following command:

```
>SET PATH=%PATH%;C:\Program Files\IBM\Java50\bin
```

Then to validate that the PATH is correct issue the following java command:

```
> java -version  
java version "1.5.0"  
Java(TM) 2 Runtime Environment, Standard Edition (build pwi32dev-  
20070511 (SR5))  
...*
```

4. By default ctgping pings the Gateway daemon only. This does not require PONG to be installed on the CICS Server. To permit testing of CICS server connectivity (using the '-c=server' option) compile the sample PONG COBOL application and deploy it in the target CICS region.

5. Channels and containers support require that the sample PONGLONG COBOL application is compiled and deployed in the target CICS region.

6. When connection to a Gateway on z/OS you are recommended to specify at least one 'DFHJVSYSTEM' environment variable parameter. DFHJVSYSTEM is required when:
 - the '-c' parameter is specified without a value and a request is sent to the default server
 - '-c' is not specified and a list systems call is run on the Gateway daemon

Usage:

Start ctgping using the supplied ctgping.bat file (Windows), or ctgping shell script (UNIX, Linux or USS). The following input parameters can be supplied:

target_GatewayURL[:port]	Gateway URL in form tcp://hostname:port
-c=CICS server	Server as defined in the CICS TG configuration file
-d=delay	The delay between iterations in seconds
-i=iterations	Number of times to execute a ping
-r=requests	Number of ECI requests sent to CICS per ping
-p=password	Password if required for authentication by the CICS TG
-u=userid	User ID required for authentication and authorization
-l=length[k]	Length of data to send to CICS

If -c=server is specified a ping request is sent through the Gateway daemon to the specified CICS server. If -c is specified without a server the request will be sent to the default server for the CICS TG. If -c is not specified the Gateway daemon is pinged using a ListSystems request.

The list systems request has a very similar code path within the CICS TG to an actual request, but does not send any data to CICS. Running ctgping with and without -c might help determine whether a problem is between the application and Gateway daemon, or between the Gateway daemon and CICS. It can also be useful to compare response times with and without -c, to get a rough sizing of how much time is taken in the Gateway side and how much is in the CICS side.

NOTE: The default server is specified in the Gateway daemon configuration. It is specified using the DFHJVSYSTEM_00 environment variable on the Gateway on z/OS. It is specified as the first SERVER section in the initialization file on distributed platforms. A list systems request to a Gateway on z/OS requires at least one DFHJVSYSTEM environment variable is required to complete successfully.

The -l=length parameter allows the length of data to be sent to, and received from, CICS to be specified. Length is a numeric value. If the letter 'k' is added to the end of the value the length is specified in kilobytes. If the value is not suffixed with a letter 'k' it is specified in bytes. For example '-l=10k' specifies that 10 kilobytes is sent on each request and '-l=100' specifies that 100 bytes is sent on each request.

When the length is less than 32500 bytes, COMMAREA based requests are used to send and receive the data. When the length is >32500 bytes channels and containers are used.

NOTE: Channels and containers require the IPIC protocol is used to communicate with CICS.

Examples:

1. `ctgping tcp://localhost:2006 -c -l=30k`
This pings the default Gateway daemon listening on the local machine and sends requests to the default CICS server. Each request sends and receives 30 kilobytes of COMMAREA data.
2. `ctgping local:// -c=tcp://myhost:6001#NETID.MYCICS -l=50k`
This pings a local gateway running on the current machine, and sends requests over an IPIC connection to the CICS server known as MYCICS in the NETID network listening on the port 6001. A 50-kilobyte container is sent and received on each request.
3. `ctgping tcp://myctg.ibm.com -r=1`
This pings a Gateway daemon running on the host "myctg" listening on the default port of 2006, the default of 5 iterations will be sent with 1 ListSystem request to the Gateway daemon per iteration.
4. `ctgping tcp://myctg.ibm.com:2007 -c=mycics -u=bob -p=password`
This pings a Gateway daemon running on the host myctg.ibm.com listening on port 2007, and sends requests to the CICS server known as "mycics" with the user ID "bob" and the password of "password"
5. `ctgping tcp://myctg.ibm.com -r=1 -i=360 -d=10`
This pings a Gateway daemon running on the host "myctg", one ECI request is sent per ping, 360 requests are sent with a delay of 10 seconds between each ping.
6. `ctgping tcp://localhost: -c`
This pings a Gateway daemon on the current machine, and sends requests to the default CICS server as defined to the CICS TG.
7. `ctgping local: -c`
This pings a local gateway running on the current machine, and sends requests to the default CICS server as defined to the CICS TG.
8. `ctgping local:// -c=tcp://myhost:6001#NETID.MYCICS`
This pings a local gateway running on the current machine, and sends requests over an IPIC connection to the CICS server known as MYCICS in the NETID network listening on the port 6001.

Usage notes:

- Response times reported are in milli-seconds as returned by the operating system clock, on certain operating systems the clock is not accurate below 10ms, and so values of 0ms should be understood as meaning <10ms.
- Open times are the time taken to create a socket from the Java client to the Gateway daemon, and to perform the CICS TG 3-way handshake. This is performed using the `JavaGateway.open()` method
- Request times to CICS are the time taken to flow an ECI request to a CICS application and to receive the response back. The CICS program specified is the sample PONG application provided with this SupportPac, although any application that can use a 140byte COMMAREA could be used.
- If `-c` is not specified the Gateway daemon process is pinged using a ListSystems requests. This allows CICS server response time issues to be isolated from general Gateway daemon response time issues.
- Close flows are the time taken to close the socket connection from the Java client to the Gateway daemon. This is measured using the `JavaGateway.close()` method.

Return codes

When running on a distributed platform or under Unix System Services on z/OS the `ctgping` utility returns the following return codes on completion:

```
RC_OK = 0;  
INVALID_PARAMETERS = 4;  
OPEN_FAIL = 8;  
RC_REQUEST_FAIL = 12;  
CLOSE_FAIL = 16;
```

Return codes are not available when `ctgping` is invoked via the sample JCL. `ctgping` is invoked by the CTGBATCH EXEC in the PING step in `ctgping.jcl`. This step does not set a return code if `ctgping` fails.

Runtime errors

1. If you receive the following message:

```
The Java class is not found: com.ibm.ctg.client.GatewayRequest
```

This indicates that you have not correctly set your CLASSPATH variable to point to the `ctgclient.jar`

2. If you receive the following message:

```
The java class is not found: com.ibm.ctg.ping.CTGPing
```

This indicates that the `ctgping.jar` file can not be accessed, possibly due to an incorrect CLASSPATH setting or due to permission or corruption.

3. If you receive the following error:

```
./ctgping: line 2: java: command not found
```

This indicates that the PATH variable does not point to a valid Java runtime environment.

4 If you receive the following error when running ctgping.jcl:

```
CTG0804E CTGBATCH Error 111 occurred starting child process for  
<FILENAME>. errno2=186319135.
```

This indicates that the HFS filename of the ctgping utility has been incorrectly specified.

5 If you receive the following error when specifying a length >32500 bytes

```
Request error "ECI_ERR_INVALID_CALL_TYPE"
```

This indicates that ctgping is attempting to use channels and containers and a protocol other than IPIC is being used to communicate with CICS. When the request size is greater than 32500 bytes, ctgping uses channels and containers, which require the IPIC protocol. Reconfigure the Gateway daemon to use IPIC.

6 If you receive the following error when -c is not specified :

```
ECI_ERR_NO_SYSTEMS
```

This indicates that no CICS systems have been defined on the Gateway daemon on z/OS. The environment variables file that the Gateway uses should be updated to include at least one DFHJVSYSTEM definition.

Sample output:

```
>ctgping.bat tcp://myctg.ibm.com -r=1 -c -l=200

Pinging gateway url: tcp://myctg.ibm.com
CICS request with 200 byte COMMEARA
Reply from tcp://myctg.ibm.com open=10ms, request=13ms, close=2ms
Reply from tcp://myctg.ibm.com open=3ms, request=7ms, close=1ms
Reply from tcp://myctg.ibm.com open=4ms, request=8ms, close=2ms
Reply from tcp://myctg.ibm.com open=3ms, request=8ms, close=1ms
Reply from tcp://myctg.ibm.com open=3ms, request=11ms, close=1ms

----ctgping statistics----
Opens issued=5, min=3ms, max=10ms, avg=4.6ms, errors=0
Requests issued=5, min=7ms, max=13ms, avg=9.4ms, errors=0
Closes issued=5, min=1ms, max=2ms, avg=1.4ms, errors=0
```

Understanding the output:

In a high-bandwidth local area network with a well-tuned Gateway daemon and CICS system response times should be <10ms. The following guidelines may help you with problem determination:

- High open values indicate either a slow network connection into the Gateway daemon or a slow Gateway daemon, possibly due to insufficient connection managers
- High close values indicate a slow network or a slow Gateway daemon, possibly due to heavy load.
- High request values but low open values indicate a slow CICS system or CICS network connection, or a constrained Gateway daemon. The number of Gateway daemon worker threads should be verified to ensure there is no queuing for worker threads, and the CICS transaction response time figures and network bandwidth should be closely analysed.