# Exercise 2

## Introduction

The script created for exercise 2 produces a circle of 6 Queue managers (QMGRs) and can pass messages either clockwise of anti-clockwise around the circle, starting at any QMGR and ending at any other QMGR in the loop.

To make it more challenging the script was extended so it can take any number of QMGRs as a parameter and create a circle of that size.

A second version of the script has been created that not only passes messages either clockwise or anti-clockwise but can also take the shortest route to the target QMGR.

## The Scripts

**circle6.sh** - creates the circle of 6 QMGRs. with queues going clockwise (forwards) or anti-clockwise (backwards).

optionally you can pass a number as a parameter and it will create a circle with that number of QMGRs instead.

**Each Queue Manager will have the artefacts listed in the table below:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object** | **Type** | **Name** | **Notes** |
| Local Queue |  | Tn | (The target queue) |
| Local Queue | XMIT | Tn.Tn+1 | For forwards channel |
| Local Queue | XMIT | Tn.Tn-1 | For backwards channel |
| listener | TCP/IP |  | Port 1540+n |
| Channel | snd | Tn.Tn+1 | Sends forwards |
| Channel | snd | Tn.Tn-1 | Sends backwards |
| Channel | rcvr | Tn+1.Tn | Receives from next |
| Channel | rcvr | Tn-1.Tn | Receives from previous |
| Remote Queue |  | T1 | Forwards (Clockwise) |
| Remote Queue |  | T2 | Forwards (Clockwise) |
| Remote Queue |  | T3 | Forwards (Clockwise) |
| Remote Queue |  | T4 | Forwards (Clockwise) |
| Remote Queue |  | T5 | Forwards (Clockwise) |
| Remote Queue |  | T6 | Forwards (Clockwise) |
| Remote Queue |  | BT1 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT2 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT3 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT4 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT5 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT6 | Backwards (Anti-clockwise) |

**NOTE: a remote queue is not added where it targets the QMGR of the same number (e.g. TEST3 does not have remote queues T3 or BT3). If more than 6 QMGRs are created additional remote queues are also created.**

**test6.sh** - Puts a 1 line message on each QMGR specifying the QMGR it started from and the target queue. It targets all the other QMGRs, in both directions. It then lists the number of messages on the local queues.  (For 6 QMGRs we expect each local Q to have 20 messages)

**circle6short.sh** - creates the same circle as circle6.sh but with the addition of a set of queues that take the shortest route to the target.

The number of QMGRs can be varied by parameter

**Each Queue Manager has the artefacts listed in the table below:**

|  |  |  |  |
| --- | --- | --- | --- |
| **Object** | **Type** | **Name** | **Notes** |
| Local Queue |  | Tn |  |
| Local Queue | XMIT | Tn.Tn+1 | For forwards channel |
| Local Queue | XMIT | Tn.Tn-1 | For backwards channel |
| listener | TCP/IP |  | Port 1540+n |
| Channel | snd | Tn.Tn+1 | Sends forwards |
| Channel | snd | Tn.Tn-1 | Sends backwards |
| Channel | rcvr | Tn+1.Tn | Receives from next |
| Channel | rcvr | Tn-1.Tn | Receives from previous |
| Remote Queue |  | T1 | Shortest route |
| Remote Queue |  | T2 | Shortest route |
| Remote Queue |  | T3 | Shortest route |
| Remote Queue |  | T4 | Shortest route |
| Remote Queue |  | T5 | Shortest route |
| Remote Queue |  | T6 | Shortest route |
| Remote Queue |  | BT1 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT2 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT3 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT4 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT5 | Backwards (Anti-clockwise) |
| Remote Queue |  | BT6 | Backwards (Anti-clockwise) |
| Remote Queue |  | FT1 | Forwards (Clockwise) |
| Remote Queue |  | FT2 | Forwards (Clockwise) |
| Remote Queue |  | FT3 | Forwards (Clockwise) |
| Remote Queue |  | FT4 | Forwards (Clockwise) |
| Remote Queue |  | FT5 | Forwards (Clockwise) |
| Remote Queue |  | FT6 | Forwards (Clockwise) |

**NOTE: a remote Queue is not added where it targets the QMGR of the same number (e.g. TEST3 does not have RemoteQ T3, FT3 or BT3). If more than 6 QMGRs are created additional remote Queues are also created.**

**test6short.sh** - tests the configuration with the extra Q's (expect to get 30 messages in each Q with 6 QMGRs)

Some utility scripts

**check6.sh** - check that the channels are all running

**clearQ.sh** – It keeps trying to clear the local target queue until it is successful. (Requires the QMGR number as a parameter)

**clearQall.sh** - call clearQ.sh for all QMGRs in parallel

**teardown.sh** - remove the QMGRs

NOTE: apart from clearQ.sh all the scripts can take the number of QMGRs as a parameter or will default to 6.