CICS Multi Tasking under Mainframe Express

Download Application

This article demonstrates how to set up Distributed Program Linking between two CICS regions running under Mainframe Express (MFE) running on the same machine. *Note: this assumes you are familiar with MFE and CICS.*

The same procedure can be used on two separate networked machines, however both machines must be run in a multi-tasking environment using the same network protocol.

To prepare you for the information contained in this article, it is suggested that you read, or be familiar with, the 'Mainframe Express Administrator's Guide, Chapter 7: CICS Installation Verification for Communications'.

This article shows how two CICS regions can talk to each other using TCPIP. The article will refer to two small MFE projects -*REGION1*and*REGION2*. See 1 MFE projects below.

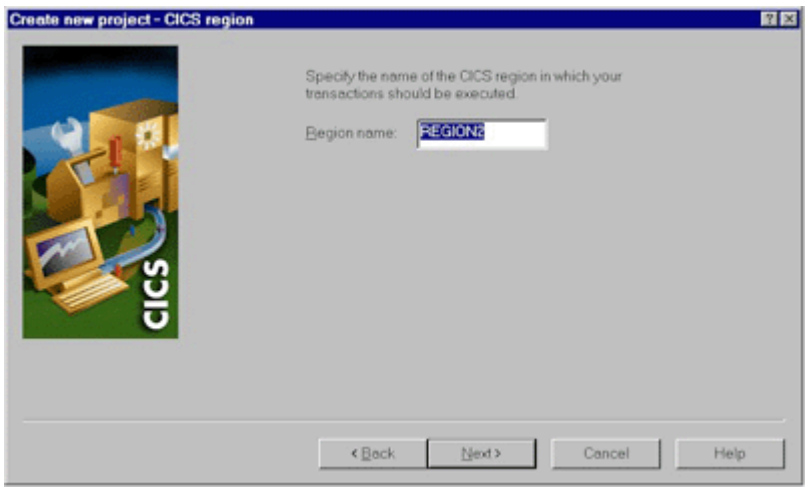
This article, also, contains references to project names, system ids and connections - the choice of names for these is up to you and your company's standards. CICS will control the process of finding the required regions.

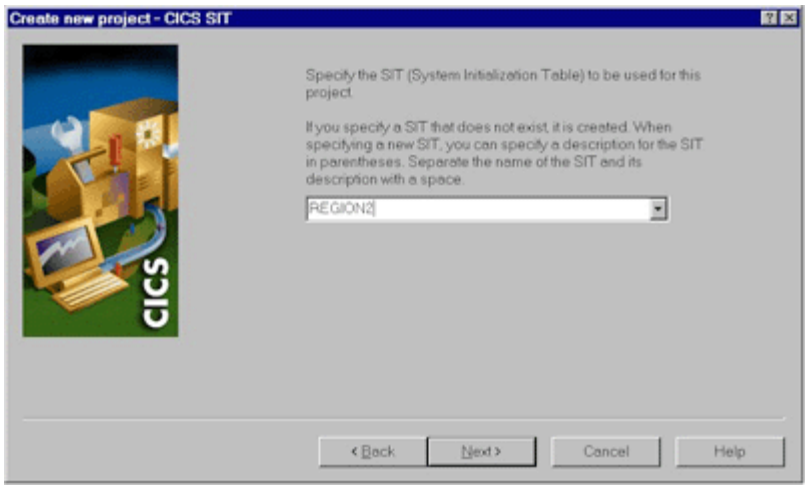
In this example, in theory only one region needs know about the other. Each CICS region needs to be aware of other region(s) required, by connection or system identification. *REGION1*needs to talk to*REGION2*and*REGION2*needs to talk to*REGION1.* This article demonstrates how to debug from one project to another project (using a different CICS region), returning to the original project.

1. MFE projects

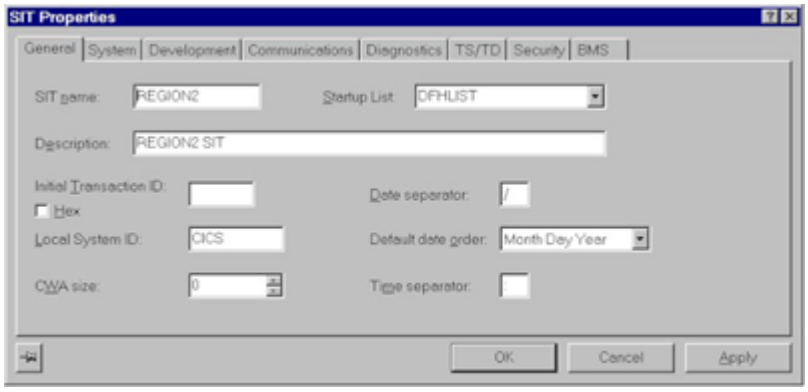
1.1. Project REGION2:
This project is the target of Distributed Program Link (DPL) from project REGION1.
The project contains program PROG2.cbl. See Appendix B for contents of PROG2.CBL

- Start MFE
- Create new project REGION2. Project uses CICS. The region name is important, in this case it should be REGION2.

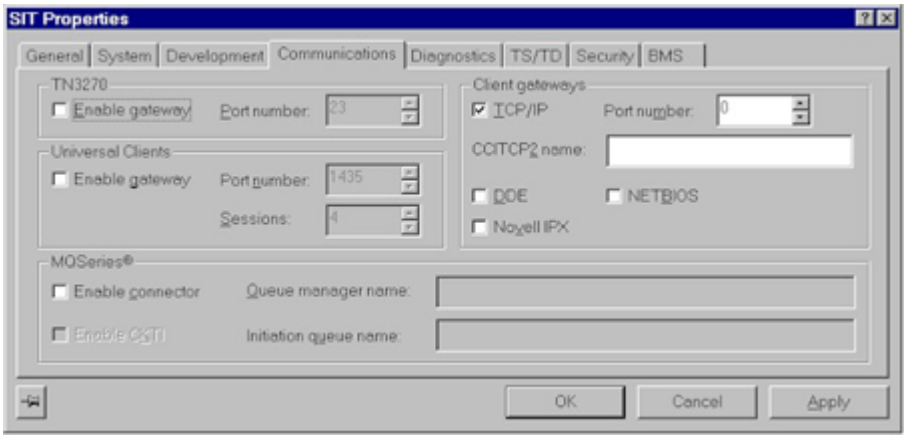




- Add COBOL program PROG2.cbl
- Compile program PROG2, using default build settings
- NOTE Project*REGION2*, CICS tab requires no specific PCT or PPT entries.
- Check System Initialization Table (SIT)
  -
  - Use CICS tab in project.
  - Highlight System Initialization tab (SIT)
  - Right click on SIT being used in right hand window. Look at the properties. The general tab should contain the CWA size, in this case 0.
    Screen print of SIT properties in project*REGION2*, general tab:



- Communications tab - client gateway TCPIP should be ticked. Screen print of SIT properties in project*REGION2*, communication tab:



1.2. Project*REGION1*(prime project)
The project contains program PROG1.cbl. See Appendix A for contents of PROG1.CBL. PROG1 contains a link to PROG2, that is it contains:

EXEC CICS LINK

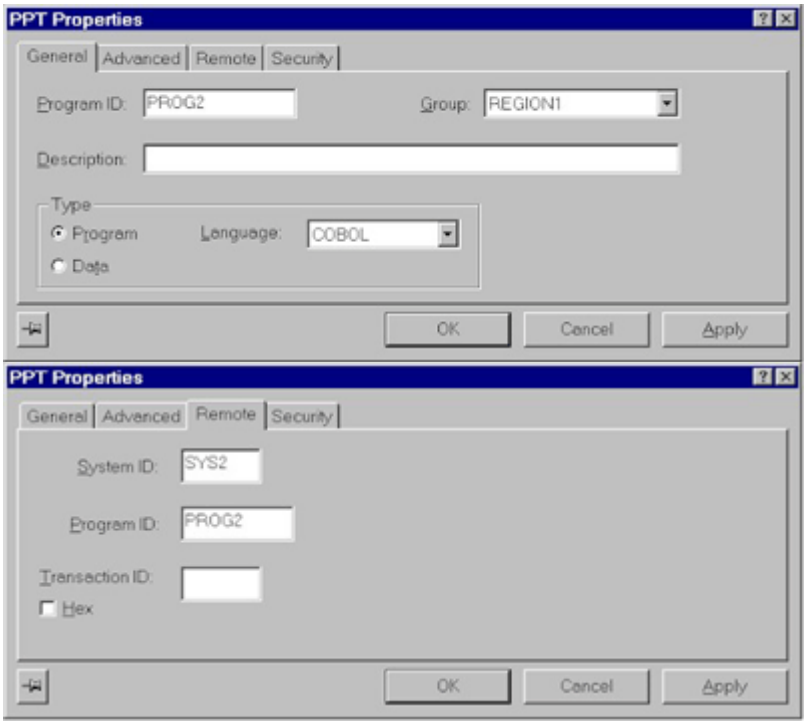
PROGRAM('PROG2')
COMMAREA(BW-COMMAREA)
END-EXEC.

- Start**a second**MFE
- Create new project*REGION1*. Project uses CICS. The region name is important; in this case it should be*REGION1*.

- Add COBOL program PROG1.cbl
- Compile program PROG1, using default build settings
- Within the project go to the CICS tab. You will need the following:

PCT entry for transaction to start PROG1, called TEST.

PPT entry for PROG2. Program id = PROG2 and Remote tab system id =**SYS2**
Screen print of the remote tab for PPT for program PROG2 in project*REGION1*:



- Define a Connection for remote system identification using the CICS resource definition tool for*REGION1*, in this example**SYS2.**

When you create project*REGION1*, ensure you create its own SIT. (See 1.2)

Within project*REGION1*:

Start the CICS resource definition -**Tools, CICS, resource definition**

F7- group resources
F2- list. Find the*REGION1*group for this CICS session and**enter**when this is highlighted
F9- terminal / connection
F3- connection
F2- list. This show the list of system id's know to the project

SYS2 (set in PPT for program in second CICS region) should be shown.
If SYS2 is not listed, escape to previous screen
F3- add connect id SYS2

Connection type CCI
Protocol CCITCP
Session max. 004
Net name*REGION2*

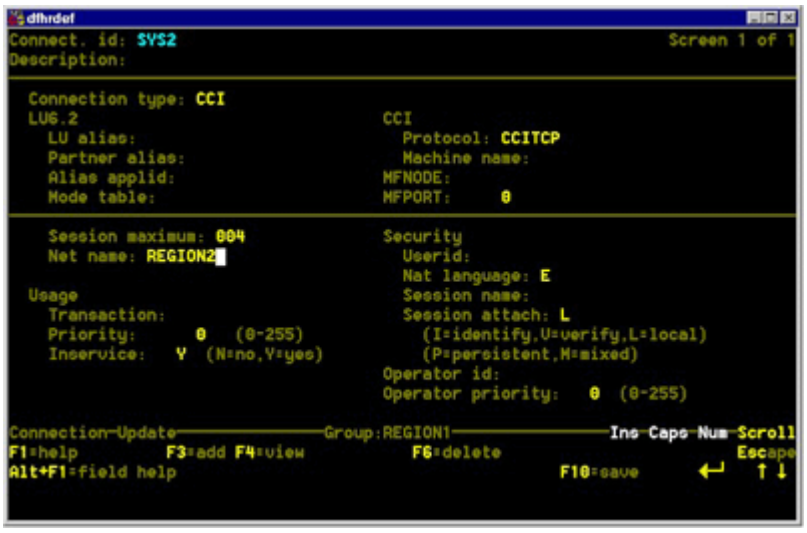
F10- Save

NOTE

It is advisable to have different names for the projects, remote id and connections. This saves confusion and makes it easier to identify any problem encountered.

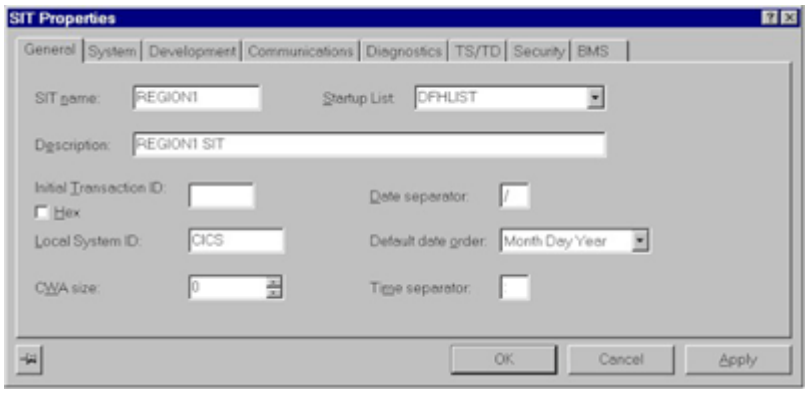
MFNODE and MFPORT may be used, but in this example we have used CCITCP2.

Screen print of connect id: SYS2 in project*REGION1*

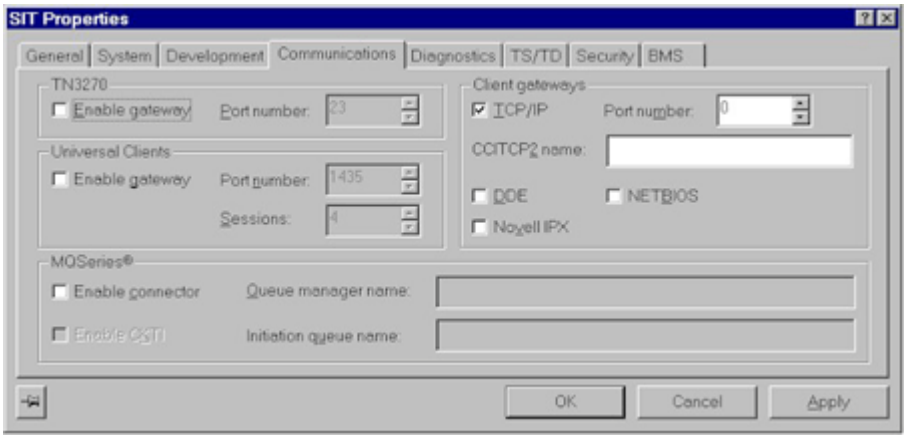


- Check System Initialization Table (SIT)

o Use CICS tab in project.
o Highlight System Initialization tab (SIT)
o Right click on SIT being used in right hand window. Look at the properties. The general tab should contain the CWA size, in this case 0.
Screen print of SIT properties in project*REGION1*, general tab:



o Communications tab - client gateway TCPIP should be ticked. Screen print of SIT properties in project*REGION1*, communication tab:



NOTE: ***From this point on you need to be running both MFE sessions. MFE session with project REGION1 and MFE session with project REGION2.***

2. To run the two regions.

NOTE: This document refers to CCITCP2, however your environment may have another configured interface on the network

2.1. Ensure CCITCP2 is running
This is only required if not using direct addressing.
At MFE command line enter Enter**CCITCP2 -D** (See 4.3)

2.2. Go to MFE session containing project*REGION2*.
Tools, CICS, Start region

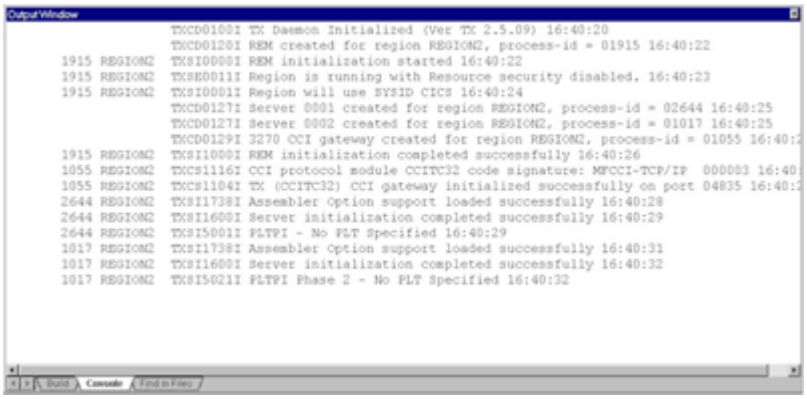
Check to ensure CICS start region successfully.
Messages will be displayed via the console in MFE, located in the bottom half of the MFE IDE.

You should see messages similar to those shown in the screen print below.
Note line containing:
' TX (CCITC32) CCI gateway initialized successfully on port 04835'.

This shows*REGION2*has successfully connected to CCITCP2 on port 04835.

If connection is not successful you will get a message like:
*REGION1*TXCS1101W CCI error: CCITC32-0016E Cannot find CCITCP2 - CCITCP cannot be used without a CCITCP2 process.

Screen print of Tools, CICS Start region for project*REGION2*.



2.3. Go to MFE session containing project*REGION1*.
Tools, CICS, Start region

Check to ensure CICS start region successfully.
You should see messages similar to those shown in the screen print below.

Note the following lines:
Line 1 'TX (CCITC32) CCI gateway initialized successfully on port 04838 16:4'
Line 2 'Attempting to establish ISC connection SYS2 to system*REGION2*16:46:0'

Line 1 shows successful connection to CCITCP2 using port 04838 for region*REGION1*
Line 2 shows connection interface - SYS2. This is also the remote id used in PPT entry of PROG2.

If you try to start*REGION1*when you have not started*REGION2*you will get the following message:
2498*REGION1*TXCS1107W Activate for connection to system SYS2 failed, reason = 0001 14:16:21

Screen print of Tools, CICS Start region for project*REGION1*

2.4. Return to the MFE session containing project*REGION2*.
Look at the messages displayed in the console log again.

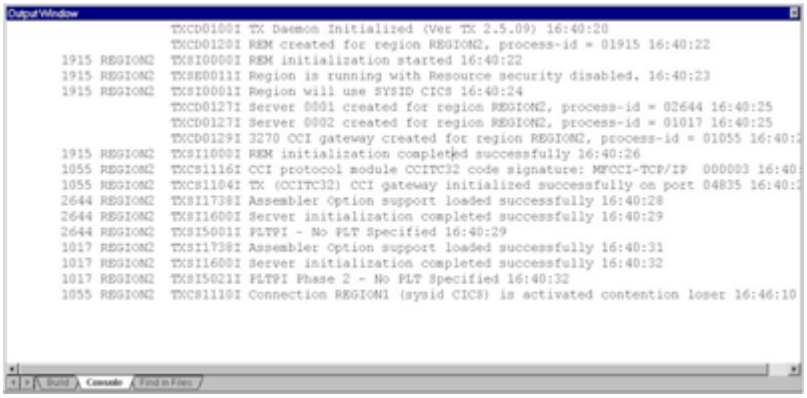
Note the last lines contains:

Connection*REGION1*(sysid SYS1) is activated contention loser 16:46:10

This shows a connection has been made between the 2 projects.

Note - 'Winner' and 'loser' are terms used by the CICS option.

Connection has to be made within a certain time limit, otherwise connection will not be found.



3.**At this stage CCITCP2 is running and both projects have their CICS region started.**

3.1. To step through PROG1 and execute PROG2.

Project*REGION1*- start debugging.
NOTE: this shows Multi-tasking debug option; you can leave this as generic.

Tools, CICS, start 3270 terminal.
Enter TEST, this is the transaction (PCT entry) in project*REGION1*to start program PROG1.

Currently you should have the following:

- MFE session with project*REGION1*and CICS region*REGION1*running.
- MFE session with project*REGION2*and CICS region*REGION2*running.
- Debug session for*REGION1*.
- Started 3270 terminal from project*REGION1*.
- Enter TEST in the 3270 terminal - this will activate the debug session of MFE project*REGION1*.

NOTE PROG2 is executed and control is passed back to PROG1. At this stage you can not step through PROG2.

3.2. To step through both PROG1 and PROG2.

Project*REGION1*- start debugging
NOTE this shows Multi-tasking debug option. For this test we will use generic.

Project*REGION2*- start debugging
NOTE this shows Multi-tasking debug option, For this test we will use generic, same as the above.

From within project*REGION1*-**Tools, CICS, start 3270 terminal**i.e. project containing the PPT.
Enter TEST, this is the transaction (PCT entry) in project*REGION1*to start program PROG1.

This will allow you to step through both programs. as you have the following:

- MFE session with project*REGION1*and CICS region*REGION1*running.
- MFE session with project*REGION2*and CICS region*REGION2*running.

- Debug session for*REGION1*.
- Debug session for*REGION2*.
- Started 3270 terminal from project*REGION1*.
- Enter TEST in the 3270 terminal - this will activate the debug session for MFE project*REGION1*and MFE project*REGION2*.

3.3. To step through PROG2 and execute PROG1.

Project*REGION2*- start debugging
NOTE: this shows Multi-tasking debug option, you can leave this as generic.

From within project*REGION1*-**Tools, CICS, start 3270 terminal**i.e. project containing the PPT.
Enter TEST, this is the transaction (PCT entry) in project*REGION1*to start program PROG1.

Currently you have the following:

- MFE session with project*REGION1*and CICS region*REGION1*running.
- MFE session with project*REGION2*and CICS region*REGION2*running.
- Debug session for*REGION2*.
- Started 3270 terminal from project REGION1.
- Enter TEST in the 3270 terminal - this will activate the debug session for MFE project*REGION2*program PROG2.

NOTE PROG2 is debuggable and PROG1 is executed. With this setup you not step through PROG1.

3.4 When you want to stop debugging, do not forget to stop the CICS regions in both the MFE sessions.
When you end debugging you may get various messages - you do want to stop execution.

3.5 When you have finished using the Common Communication Interface, remember to return to the DOS prompt and exit the Common Communication Interface i.e. control C. (See 4.3)
Be aware if you are running this all locally on you pc, you should exit the Common Communication Interface.
If you are using a network copy this should NOT be exited.

4. Ensure you have Common Communication Interface set up correctly.

Before the CCI modules can work effectively the appropriate TCPIP network hardware and software drivers must be installed, configured and verified to be working.

4.1. Ensure both client and server (if used) are set up to communicate with the same CCI.

4.2. Check C:\winnt\system32\drivers\etc\services.
This should contain:
mfcobol 86/udp
This should have been set up using start, programs, MFE25, configuration, common communication interface.

NOTE: If this is your first time of using CCITCP2, ensure your local machine is known to CCITCP2.
MFE25, configuration, common communication interface - enter local machine name and click OK

4.3. If you are using TCPIP, you must first start a ccitcp2 daemon if one is not already running on the network.

Ensure CCITCP2 is waiting for a request (that is you have a communication connection):
At the MFE command line enter: **CCITCP2 ?**

This should provide you with CCITCP2 -0102I usages;

-v Show version information
-d Show debug information
Etc

Enter**CCITCP2 -D**- the last line should show: Waiting for request

Control c exits this request.

Useful documents in MFE bookshelf:

Configuring CCI

Master Index CICS

Mainframe Express Administrator's Guide, Chapter 7: CICS Installation for Communications

EXTRA notes to create a connection between REGION2 and REGION1.

In this example it was NOT necessary to have a connection set up between*REGION2*and*REGION1*.

These notes have been left to help create the connection if required. The example above does work if this connection is created.

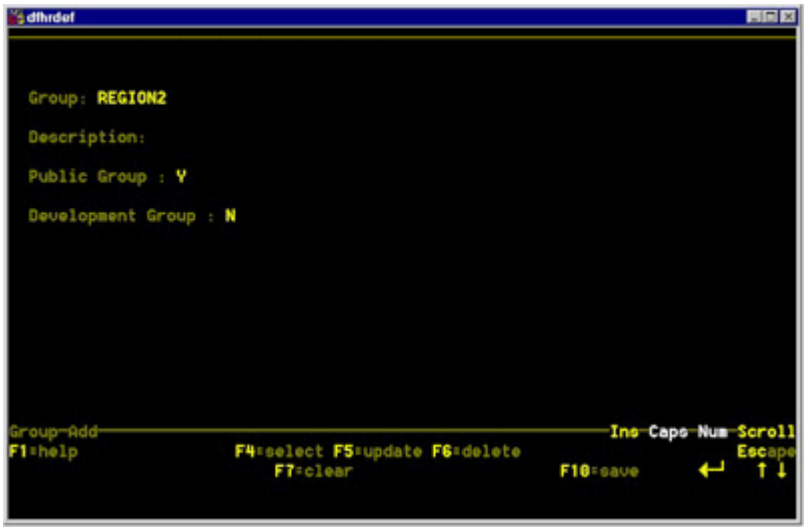
Define a Connection in the CICS resource definition tool for connection from the CICS*REGION2*to the CICS*REGION1*, in this example SYS1.

Within project*REGION2*:

Because*REGION2*contains no PCT, FCT or PPT we need to create group*REGION2*. If you do add a PCT, FCT or PPT the group will automatically be created.

Start the CICS resource definition -**Tools, CICS, resource definition:**

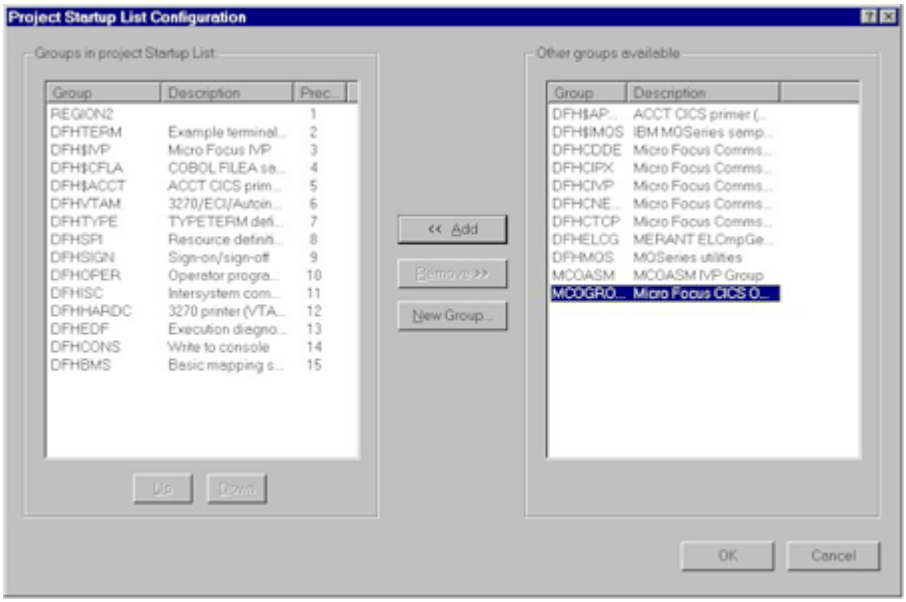
F7- group resources
type*REGION2*after group
F3- add



F10- save
Escape to exit from CICS resource definition tool

From the CICS tab:

Highlight CICS system
Right mouse click
Select Refresh CICS View from RDF
Highlight CICS system
Right mouse click
Select startup list
Add group*REGION2*to the startup list



Start the CICS resource definition -**Tools, CICS, resource definition:**

F7- group resources
F2- list. Find*REGION2*i.e.the group for this CICS session and**enter**when this is highlighted
F9- terminal / connection

F3- connection
F2- list. This shows the list of connection for the project

See if SYS1 is shown.
If SYS1 is not listed, escape to previous screen
F3- add connect id SYS1

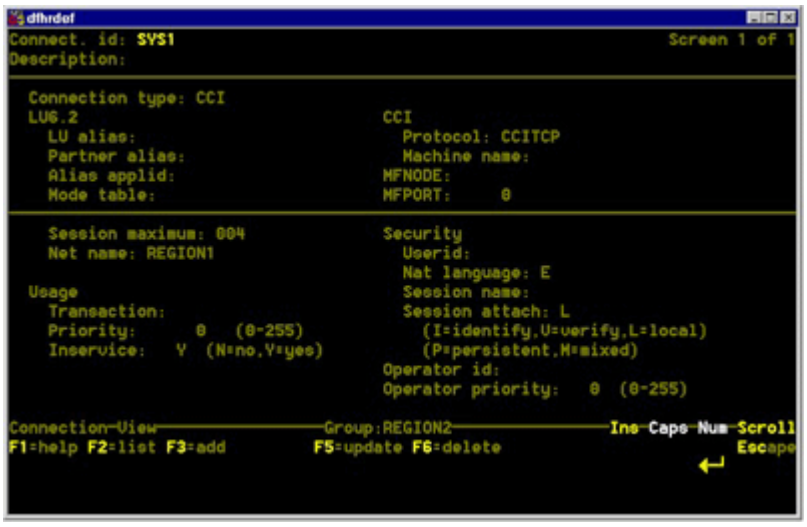
Connection type CCI
Protocol CCITCP
Session max. 004
Net name*REGION1*

F10- Save

Connect id. is SYS1, used for connection to project*REGION2*.

Net name of*REGION1*is the connection between project*REGION1*and*REGION2*

Screen print of SYS1 connection:



Appendix A

Contents of COBOL program PROG1 - PROG1.CBL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG1.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
DATA DIVISION.
*
WORKING-STORAGE SECTION.
01 BW-COMMAREA PIC X(100).
LINKAGE SECTION.
PROCEDURE DIVISION.
```

```
MOVE 'THIS IS FOR MULTI TASK TEST' TO BW-COMMAREA.
EXEC CICS LINK

PROGRAM('PROG2')
COMMAREA(BW-COMMAREA)

END-EXEC.

*

EXEC CICS RETURN
END-EXEC.
STOP RUN.
```

Appendix B

Contents of COBOL program PROG2 - PROG2.CBL

```
IDENTIFICATION DIVISION.
PROGRAM-ID. PROG2.
ENVIRONMENT DIVISION.
INPUT-OUTPUT SECTION.
DATA DIVISION.
*
WORKING-STORAGE SECTION.
01 WORK-AREA PIC X(100).
LINKAGE SECTION.
01 DFHCOMMAREA.

03 OCCURS 0 TO 32767 TIMES DEPENDING ON EIBCALEN PIC X(1).

PROCEDURE DIVISION USING DFHCOMMAREA.

MOVE SPACES TO DFHCOMMAREA.
MOVE 25 TO EIBCALEN.
MOVE 'THIS SHOWS PROG2 EXECUTED' TO DFHCOMMAREA.
MOVE DFHCOMMAREA TO WORK-AREA.

*

EXEC CICS RETURN
END-EXEC.
```