EMRO Introduction

- Initial Discussion
- Primary and Secondary Bridge Facilities
- Transaction Routing
- Application Context Data
- Propagating Bridge Facility TERMIDs Across AORs
- Overriding Default TERMID/NETNAME for Bridge Facilities

Initial Discussion

CICS TS 2.2 introduced the Link3270 Bridge interface. When HostBridge invokes Link3270 Bridge (DFHL3270) to allocate a bridge facility, DFHL3270 makes a routing decision based upon the first transaction run. If the transaction is defined in a different region than DFHL3270 is running, DFHL3270 will act as a proxy and pass control to DFHL3270 in the remote region (running under a mirror transaction). From that point on, all transactions executed under that bridge facility will be routed to that region, even though subsequent transactions might not exist there.

The diagram below illustrates the operation of HostBridge and the CICS Link3270 Bridge interface when EMRO is not used.

Link3270 Bridge dictates that all transactions executed under a single bridge facility must run in the same application-owning region (AOR). In some situations, this may not be a limitation. However, this is a serious limitation for organizations with complex and highly scaled multi-region operation (MRO) environments. HostBridge overcomes these limitations through a collection of features referred to as Enhanced MRO (EMRO).

HostBridge EMRO provides the following benefits:

- HostBridge can run in its own region while transactions can execute in multiple AORs.
- Distributed programs that invoke HostBridge do not have to know where a transaction exists.
- EMRO manages application context data across regions automatically, just as CICS does for terminal facilities.
- EMRO is transparent to the applications that invoke HostBridge.

The diagram below illustrates the operation of HostBridge and the CICS Link3270 Bridge interface when EMRO is used.
Primary and Secondary Bridge Facilities

HostBridge EMRO makes a distinction between a "primary bridge facility" and a "secondary bridge facility". The "primary" bridge facility is the one created by HostBridge in the region in which HostBridge runs. A "secondary" bridge facility is one created by HostBridge for running a transaction in a different region.

When running terminal oriented transactions via HostBridge, there will always be one primary bridge facility. Secondary bridge facilities are automatically created and destroyed as HostBridge needs them to run transactions in other regions. HostBridge manages multiple bridge facilities under a single HostBridge session represented by a HostBridge "token".

Transaction Routing

HostBridge EMRO uses a transaction routing exit to determine where a transaction exists and returns the name of the region to HostBridge. The HB_ROUTING_EXIT directive determines whether HostBridge LINKs to the specified routing exit program.

After the routing exit returns control, HostBridge LINKs to DFHL3270 in the region indicated by the routing exit. As a result, DFHL3270 receives control in the region in which the transaction exists. This process circumvents the limitation in DFHL3270's routing behavior.

HostBridge includes a default routing exit (HBR$RTEX). This exit issues an EXEC CICS INQUIRE command to determine the region in which the transaction exists. The transaction routing exit can be customized and tailored to meet unique customer requirements (e.g., CICSPlex SM support). Typically, HostBridge works with customers to perform this customization as part of the EMRO implementation process.

Application Context Data

For transactions that run under a terminal facility (as opposed to a bridge facility), CICS automatically moves application context data between regions. For example, assume the following:

- Terminals are owned by region TOR1
- The TERMTYPE associated with the terminals indicated that a TCTUA is to be allocated for the terminal facility.
- Transaction A is located in AOR1
1. When the terminal logs on in TOR1, CICS creates a terminal facility and allocates the TCTUA in TOR1.
2. When the end user runs transaction A, CICS determines that A exists in AOR1. Prior to starting A in AOR1, CICS copies the terminal facility from TOR1 to AOR1. As part of this process, CICS will copy the TCTUA data from TOR1 to AOR1. As a result, transaction A runs with the correct application context data. Because transaction A terminates with RETURN TRANSID('B') COMMAREA(...), CICS automatically copies the current TCTUA and COMMAREA data from AOR1 back to TOR1.
3. Once the end user responds with input, transaction B receives control and CICS determines that B exists in AOR2. Prior to starting B in AOR2, CICS copies the terminal facility from TOR1 to AOR2. As part of this process, CICS copies the TCTUA and COMMAREA data from TOR1 to AOR2. As a result, Transaction B runs with the correct application context data available.

For transactions that run under a bridge facility, CICS TS does not automatically copy application context data and does not support transaction routing for bridge facilities. HostBridge EMRO overcomes these limitations by using a service transaction (HBGP) to move application context data between regions (across bridge facilities).

The HostBridge HB_CONTEXT_TRAN directive controls the use of HBGP. HostBridge management of application context data is completely transparent to the CICS transactions being executed.

### Propagating Bridge Facility TERMIDs Across AORs

Most applications that exploit CICS MRO capabilities assume that the same terminal ID (termid) is used for all transactions run under a given session, regardless of the region in which they run. For example, the transactions associated with a particular application might share data using a TS Queue whose name is based upon the termid.

Given the default behavior of CICS, this assumption may be violated when running transactions under a bridge facility. Here's why:

- **By default, the termid/netname generated by CICS for a bridge facility has a right brace (}) at the end (as its fourth character).**
- **For a program such as HostBridge to propagate the termid/netname from its region to an AOR, the program must specify the termid/netname as a parameter when it creates the bridge facility used to run the transaction in the AOR. This termid/netname is referred to as a "user defined" termid/netname.**
- **The CICS Link3270 Bridge mechanism enforces the following character set rules for "user defined" termids and netnames:**
  - The character set allowed for a user defined termid is "A-Z a-z 0-9 @#./%_&?!:|", plus the cent sign.
  - The character set allowed for a user defined netname is more restrictive: "A-Z 0-9 $@#".

As you can see, the right brace is not included in either of these character sets. Given the default behavior of CICS, when HostBridge creates a secondary bridge facility it checks the formulation of the termid associated with the primary bridge facility:

- **If the termid does end with a right brace, HostBridge will not specify the termid/netname of the primary bridge facility as the "user defined" termid/netname (because it would be invalid). As a result, the Link3270 Bridge mechanism will generate a different termid/netname for the bridge facility used to run transactions in the other region.**
- **If the termid does not end with a right brace, HostBridge will specify the termid/netname of the primary bridge facility as the "user defined" termid/netname. As a result, the Link3270 Bridge mechanism will use the same termid/netname for the bridge facility used to run transactions in the other region.**

As a result, if the CICS applications you will be running via HostBridge assume that the same termid will be used for all transactions run under a given session (regardless of the region in which they are run), then we must ensure that the termid/netname of the primary bridge facility is "MRO safe".

### Overriding Default TERMID/NETNAME for Bridge Facilities

HostBridge overrides the assignment of the termid when it creates the primary bridge facility. Rather than using a right brace as the rightmost character of a termid, Hostbridge uses $, @, or # (since netname is usually derived from termid, Hostbridge assumes that the only "MRO safe" characters are: A-Z 9-0 $ @ and #).

The override takes place in functions 15 and 17 in DFHZATDX, the CICS "terminal autoinstall user program". A sample of the code that is required to use @ as the fourth character of bridge facility termids and netnames may be found here.

If your CICS applications assume that the same termid is used for all transactions run under a given session (regardless of the region in which they are run), then you must implement these changes (or equivalent) in your autoinstall exit.

Please note the following:

- AIBRIDGE=YES must be specified in the SIT, but only in the regions in which HostBridge runs; this will cause DFHZATDX to be invoked for bridge facilities.
- This autoinstall exit only needs to be implemented in the region(s) in which Hostbridge runs; it does not need to be implemented in the regions in which HostBridge starts transactions (i.e., the AORs).