General Information

This article contains information about managing state and sessions, and using HostBridge user exits.

Table of Contents

- Managing State and Sessions
  - Beginning a New “session”
  - Using an Existing “session”
  - Controlling Bridge Facility Timeouts and Cleanup
- Password Change Using HTTP Headers
- User Exit Programs
  - Initialization and TidyUp Exits
  - HBR$FAIN
  - HBR$FATU
  - HBR$AUTH
- Specifying Alias Transactions
- START BREXIT and Link3270 Bridge in Transaction Server 2.2
  - Naming Bridge Facilities
- Multiregion Operation (MRO)
- Executing Terminal-Oriented Transactions that Use DB2
- Load Balancing and HTTP Headers

Managing State and Sessions

The following discussion regarding state and session management is only relevant when using HostBridge to access terminal oriented transactions.

When a user at a 3270 terminal (or emulator) logs on to CICS, two things happen:

- External to CICS, a persistent connection is created between the terminal and CICS
- Internally, CICS creates a session for the user

These are two different and distinct concepts. Within CICS, the user’s session is represented by an area of memory called the “principal facility”. When the user logs off, the session times out, or the connection with the terminal is broken, the CICS session ends and CICS deletes the principal facility. The connection between the terminal and CICS exists until the session ends.

When a remote application uses HostBridge to interact with a CICS terminal-oriented transaction, things work differently.

First, there is no persistent connection between the remote application and HostBridge. Instead, regardless of whether you use the HTTP or LINK interface, a stateless connection exists between the remote application and HostBridge. HTTP servers have no inherent mechanism to allow them to associate one transaction with another, so the concept of a session does not exist. The same holds true when one program invokes another using a LINK-style interface. As a result, HostBridge and the remote application must agree upon a mechanism to achieve a “virtual” persistent connection.

Secondly, when accessing terminal-oriented applications, HostBridge uses the 3270 bridge interface. The Bridge interface makes it possible for a software component, such as HostBridge, to intercept the flow of data into, and out of, a CICS transaction. The concept of a CICS “session” still exists, but it uses a “bridge facility” instead of a principal facility. A bridge facility exists until either the program using the Bridge interface (i.e., HostBridge) requests that it be terminated or the bridge facility times out (has been inactive for a given period of time).

HostBridge uses a token passing mechanism to allow a remote application to execute multiple transactions under a single session (i.e., bridge facility) and maintain a “virtual” persistent connection between the remote application and HostBridge.

Beginning a New “session”

When HostBridge receives a transaction request, it examines the request for the HB_TOKEN command. If this command is not present, or if HB_TOKEN=0, HostBridge assumes that this is the first transaction of a new session and causes a new bridge facility to be created within CICS. HostBridge also generates an alphanumeric token value for this session. The token serves as the external identifier for a particular session and Bridge facility. HostBridge uses the token to manage various areas of storage that it allocates for each session. HostBridge includes the token in the resulting XML document sent back to the requesting application. The requesting application must return this token value back to HostBridge on subsequent requests to continue executing transactions within the context of this session.

Using an Existing “session”

When HostBridge receives a request that includes the HB_TOKEN command, it associates the request with the corresponding bridge facility. This allows a remote application to tell HostBridge to continue executing transactions within the context of a previously established session.

When you reuse a session for a pseudo-conversational transaction HostBridge uses the NEXT TRANID specified by the CICS application. If you specify HB_TRANID, HostBridge will ignore the command and execute the transaction specified as the NEXT TRANID from the previous request. Thus, there is no need to specify HB_TRANID and HB_TOKEN in the same request.
When a conversational transaction returns control to CICS it does not specify the name of the next transaction to be executed. Thus, HB_TRANID will be ignored until the conversational program finishes (task end).

Controlling Bridge Facility Timeouts and Cleanup

A bridge facility exists until either HostBridge requests that it be terminated or the bridge facility is inactive for a specified period and times out. When a bridge facility is inactive for this period, CICS terminates the facility and notifies HostBridge of the event. HostBridge then expires the associated token and deletes the storage for the session. HostBridge returns an error message if it receives a request that references an expired token.

The default timeout value is 300 seconds. HostBridge allows remote applications to control the timeout value using the HB_FACILITY_TIMEOUT command. Note that specifying HB_FACILITY_TIMEOUT=0 implicitly instructs HostBridge to request that CICS terminate the bridge facility immediately.

Under CICS TS 2.2 and the Link3270 interface, you can instruct HostBridge to destroy a facility immediately after a request finishes. The facility is immediately cleaned up and does not have to wait on a timeout event. To instruct HostBridge to destroy the facility as soon as a request finishes, add HB_DELETE_SESSION=1 to your request. After HostBridge returns the XML from the transaction, the token will no longer be valid for the destroyed facility.

Password Change Using HTTP Headers

For customers who need to be able to effect a password change from the middle tier, HostBridge sends two extra HTTP headers with numeric codes to identify authorization status:

`HB-AUTH: [NUMBER]`
`HB-AUTH2: [NUMBER]`

Values for the headers appear below.

<table>
<thead>
<tr>
<th>HBAUTH value</th>
<th>HBAUTH2 value</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 - Normal</td>
<td>0 - OK</td>
</tr>
<tr>
<td>4 - Invalid input</td>
<td>8 - Credentials not specified</td>
</tr>
<tr>
<td>12 - Invalid request</td>
<td>13 - There is an unknown return code in ESMRESP from the ESM</td>
</tr>
<tr>
<td></td>
<td>18 - The CICS ESM interface is not initialized</td>
</tr>
<tr>
<td></td>
<td>29 - The ESM is not responding</td>
</tr>
<tr>
<td></td>
<td>32 - The userid field contains a blank in an invalid position</td>
</tr>
<tr>
<td>16 - Not authorized</td>
<td>2 - The supplied password is wrong</td>
</tr>
<tr>
<td></td>
<td>3 - A new password is required</td>
</tr>
<tr>
<td></td>
<td>4 - The new password is not acceptable</td>
</tr>
<tr>
<td></td>
<td>19 - The USERID is revoked</td>
</tr>
<tr>
<td></td>
<td>22 - Request failed during seclabel processing</td>
</tr>
<tr>
<td></td>
<td>31 - User revoked in default group</td>
</tr>
<tr>
<td>20 - USERID error</td>
<td>8 - The USERID is not known to the ESM</td>
</tr>
<tr>
<td>24 - Other error</td>
<td></td>
</tr>
</tbody>
</table>

Typically, HostBridge will return (0,0), which means everything is normal.

`HB-AUTH: 0`
`HB-AUTH2: 0`
If the HTTP headers return values of (16,3), which means a new password is required.

```
HB-AUTH:  16
HB-AUTH2:  3
```

To change a password,

- Rename the HostBridge file HBR#AUTH to HBR$AUTH.
- In TCPIPSERVICE, set Authenticate to “No.”

HBR$AUTH is required for you to use the password change capabilities

When a valid password change request is sent in, the user’s password is changed to the new password. Subsequent HostBridge requests must use this new password instead of the old one.

- Send in the standard Basic header with valid credentials. In addition to the standard Authenticate header, pass in:

  ```
  X-HB-NEWPWD: value
  ```

  where “value” is username:new_password (base64 encoded). This is the same format that Authenticate uses.

User Exit Programs

Customers can create their own user exit programs to tailor the behavior of HostBridge. HostBridge user exit programs should be in the same load module library as HostBridge. You must also define and install the user exit programs to CICS.

Initialization and TidyUp Exits

HBR$FAIN and HBR$FATU are the Initialization and TidyUp exits. HostBridge invokes them whenever a bridge facility is created or destroyed (respectively). You can write them in any language that CICS supports. Common uses of HBR$FAIN and HBR$FATU include implementing local security requirements, performing application or TCTUA initialization, and creating messages for logging and auditing purposes.

HBR$FAIN and HBR$FATU are the HostBridge equivalents of the standard CICS SIGNON and SIGNOFF exits. CICS invokes the SIGNON and SIGNOFF exits whenever a principal facility is created or destroyed after a terminal logs on or off. However, CICS does not invoke SIGNON and SIGNOFF for bridge facilities – only principal facilities because SIGNON/SIGNOFF assume that a terminal exists. If your installation uses the SIGNON/SIGNOFF exits, then you may need to develop the equivalent functionality in the HBR$FAIN/HBR$FATU exits.

One technique that HostBridge customers use is to make a LINKable version of their SIGNON and SIGNOFF exits. You can write HBR$FAIN and HBR$FATU as small “wrapper” programs that LINK to the SIGNON/SIGNOFF routines. However, you must insure that your SIGNON/SIGNOFF exits do not perform functions that are illegal when a principal facility does not exist. For example, if HBR$FAIN calls your SIGNON routine, and your SIGNON routine issues an EXEC CICS SIGNON command, an abend will occur because EXEC CICS SIGNON requires that a principal facility exist. To deal with this, you can modify your SIGNON/SIGNOFF routines to determine whether the context is a principal facility or a bridge facility before issuing any such commands.

```
You do not need to establish user context for HBR$FAIN as you do SIGNON. User context is established at an earlier stage in the HostBridge environment.
```

HBR$FAIN

If defined, HostBridge LINKs to HBR$FAIN immediately after a bridge facility is created. HBR$FAIN executes before the requested transaction. Prior to LINKing to HBR$FAIN, HostBridge creates a COMMAREA that contains the facility name.

**Table 6-2. HBR$FAIN program COMMAREA definition**

<table>
<thead>
<tr>
<th>Offset Hex</th>
<th>Type</th>
<th>Len</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (zero)</td>
<td>CHARACTER</td>
<td>4</td>
<td>FACILITY NAME</td>
</tr>
</tbody>
</table>

HBR$FATU

If defined, HostBridge LINKs to HBR$FATU immediately before a bridge facility is destroyed. Prior to LINKing to HBR$FATU, HostBridge creates a COMMAREA that contains the facility name.
HBR$AUTH

HBR$AUTH is a HostBridge-provided program that is only required if you have CICS TS 1.3, use the CICS HTTP Listener, and have not installed APAR PQ36169. HBR$AUTH provides basic authentication services using the EXEC CICS VERIFY command. We recommend installing APAR PQ36169 as this moves the user authentication into the CICS HTTP Listener. To use HBR$AUTH, rename the module in the HBRxxx.LOADLIB from HBR#AUTH to HBR$AUTH. Then define and install the program definition shown below.

```
DEFINE PROGRAM(HBR$AUTH) GROUP(HBRGROUP)
  LANGUAGE(ASSEMBLER)
  DATALLOCATION(ANY)
  EXECUTE(CICS)
```

HBR$AUTH is required for you to use the password change capabilities described on page 6-4. To allow password change, TCPIPSERVICE should be modified where Authenticate is set to "No."

Specifying Alias Transactions

The alias transaction is the transaction that executes the user request in the CICS environment. By default, HostBridge executes under the HWBA transaction. However, if you want to tailor HostBridge’s behavior, you can specify a transaction other than HWBA to execute HostBridge by passing the new transaction ID to the HostBridge analyzer along the HTTP command string.

```
http://ip-addr:port/hostbridge?hb_tranid=yyyy...
```

In the case above, HostBridge operates under the default HWBA transaction ID when it executes transaction yyyy. To use an alias transaction other than HWBA, you specify the ID as shown below.

```
http://ip-addr:port/hostbridge/xxxx?hb_tranid=yyyy...
```

START BREXIT and Link3270 Bridge in Transaction Server 2.2

In CICS Transaction Server 1.2, IBM introduced the 3270 bridge to facilitate access to BMS application data without requiring a physical terminal or terminal emulator. HostBridge uses the 3270 bridge to access both 3270 and BMS transactions for companies running CICS TS 1.3 or higher.

In CICS TS 2.2, the 3270 bridge exists in two forms: the START BREXIT mechanism (which is the same 3270 bridge that existed in CICS TS 1.3) and the Link3270 mechanism. The main differences between the two 3270 bridge mechanisms are terminal ID naming and multiregion operation (MRO).

Naming Bridge Facilities

The 3270 bridge that ships with CICS TS 1.3 does not provide a mechanism to HostBridge for setting or requesting a particular bridge facility. However, CICS TS 1.3 is consistent about the names that it assigns to them. Naming starts at |AAA. Each subsequent facility name is incremented by one letter. For example, after |AAA, |AAB, and |AAC will be assigned. Once a HostBridge session ends, the associated facility is returned to the pool. The lowest "lettered" facility that is available in the pool will be used first. For instance, if |AAA, |AAB, and |AAC are being used by HostBridge, and the session that was using|AAA ends, |AAA will be used for next HostBridge session that is started.

If you have an application that requires that terminals be predefined to it, you could define a range of names that would cover the maximum number of sessions that you anticipate using at one time. For instance, if you don't expect to exceed 26 concurrent sessions, you could define |AAA through |AAZ as terminals for your application.

The Link3270 Bridge included in CICS TS 2.2 provides a method for creating a pool of specified terminal IDs which you can access through HostBridge. You can either build a file of facility names in DFHBRNSF, or you can use the HB_TERMD command to specify a facility name under which a transaction should operate. Thus you can specify a facility name that matches the terminal ID expected by your application.

Multiregion Operation (MRO)
The START BREXIT mechanism available in CICS TS 1.3 or higher does not support multiregion operation (MRO). Thus, HostBridge needed to run within the same regions as the applications it would access. The Link3270 mechanism supports MRO, so HostBridge can run in one region and access terminal-oriented transactions in another.

Note that HostBridge does not need to be installed in each region where your transactions run. However, if these transactions use BMS, then the BMS map load modules must be defined and accessible to the region in which HostBridge runs.

To use HostBridge with the Link3270 bridge MRO, you must set up your transactions to work with Link 3270 as described in the CICS Resource Definition Guide, Version 2 Release 2 (SC34-5990-04).

Executing Terminal-Oriented Transactions that Use DB2

Dynamic plan switching (DPS) is a feature of DB2 for CICS that allows more than one DB2 plan in a transaction. To accomplish plan switching, DB2 threads can be released at each syncpoint and then reacquired to drive the dynamic plan exit. The dynamic plan exit can then be used to select the required plan for the program. This enables the use of a different plan for different units of work within a transaction.

For a transaction to release a thread at syncpoint, the transaction must be driven from a terminal. By default, threads are not released for transactions not driven from a terminal. This can create a problem for programs that use the 3270 bridge because, from DB2's perspective, a terminal-oriented transaction running under a bridge facility has not been driven from a terminal!

To change DB2's behavior to work correctly with HostBridge, you must specify NONTERMREL=YES on the CICS DB2CONN definition in the region in which the transaction runs. The following screen illustrates a DB2COMM resource with NONTERMREL set correctly to YES.

```
OBJECT CHARACTERISTICS
  CEDA View DB2Conn( RCT20 )
  DB2Conn : RCT20
  Group : RCT20
  Description : DB2Conn migrated from load module DMPRCT20

CONNECTION ATTRIBUTES
  Connectorerror : Sqlcode | Abend
  DB2id : TDB1
  MSGQUEUEL : CHT
  MSGQUEUEL : 
  Nontermrel : Yes | No
  Purgecycle : 00 , 30 0-59
  Sigmalid :
  STANDbymode : Connect | Reconnect | Connect | Noconnect
  STATSqueue : CSSL
  TChlimit : 0012 4-2000
  THREADerror : Abend | N906 | N906 | Abend

POOL THREAD ATTRIBUTES

SYSSID=TT20 APPLID=CICSTT20
PF 1 HELP 2 COM 3 END 6 CRSR 7 SEH 8 SFM 9 MSG 10 SB 11 SF 12 CNCL
```

**Figure 6-1.** DB2COMM resource definition with NONTERMREL=Yes

Load Balancing and HTTP Headers

HostBridge includes HTTP headers in output to aid with load balancing devices. These headers tie sessions to a particular CICS region so subsequent requests return to that region.
Table 6-4. HTTP headers used for load balancing

<table>
<thead>
<tr>
<th>Header</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-HB-ADDRESS</td>
<td>IP address of the HostBridge region</td>
</tr>
<tr>
<td>X-HB-PORT</td>
<td>Port assigned to the HostBridge region</td>
</tr>
<tr>
<td>X-HB-TOKEN</td>
<td>If a token exists,</td>
</tr>
</tbody>
</table>

These headers appear in the form shown below.

X-HB-ADDRESS: 127.0.0.1
X-HB-PORT: 03029
X-HB-TOKEN: e2bab5c