Can IBM Debug Tool and RDz Integrated Debugger both coexist in same CICS region?

With v9.1.1, the Integrated Debugger feature introduced a common adapter which made it possible for both IBM Debug Tool and Integrated Debugger to coexist. Since both are LE debuggers, they include a CEEVDBG module which registers with LE at the region start up. There can only be one debugger registered with LE so the first CEEVDBG found in the search path will determine the debugger used. For this to be possible the RD/z load library has to be in the DFHRPL placed in front of the Debug Tool SEQAMOD library.

Also, prior to Rational Developer for System z (RDz) v9.1.1, the Integrated Debugger only worked with the CADP transaction for creating debug profiles. The SIT parm of DEBUGTOOL=YES enables CADP transaction. The IBM Debug Tool uses the DTCN transaction for creating debug profiles and requires SIT parm of DEBUGTOOL=NO which are mutually exclusive to CADP. (IBM Debug Tool can use both DTCN and CADP created debug profiles). The order of the load libraries in the STEPLIB / DFHRPL determines which debugger was used prior to 9.1.1.

When a debug session is triggered, the common adapter gains control and parses the LE options to determine which debugger to call. If the directives DBM or DIRECT are detected, Integrated Debugger handles the debug session itself, otherwise the session is passed onto IBM Debug Tool.

**Integrated Debugger CICS updates:**

Assuming that IBM Debug Tool is already installed and working in your CICS region, here are a few things System Admin should note as they follow the instructions in the RDz 9.1.1 Host Configuration Guide, specifically the section titled (Optional) Integrated Debugger in Chapter 7 (Optional) Other customization tasks.

**CICS system initialization (SIT) parameter updates:**

- Contrary to what is documented for the SIT parm, specify DEBUGTOOL=NO to enable the DTCN DFHRPL:

- Only one Language Environment (LE) based debugger can be active in a given application, CICS region, DB2 stored procedure, or IMS transaction. For CICS, make sure Integrated Debugger is loaded first by placing the RDz loadlib’s are higher up in the DFHRPL concatenation list
Use DTCN for creating debug profiles

Here we cover 3 remote debug scenarios:

1. IBM Debug Tool using IP Address
2. Integrated Debugger using DIRECT directive
3. Integrated Debugger DBM directive

In preparation, grab the workstation IP address, verify that the Debug UI daemon listening and note the port number

IBM Debug Tool – IP Address:

From a CICS terminal, run the DTCN transaction to define the debug profile for debugging with IBM Debug Tool. See the examples below where I specify an * for Terminal ID because I am going to do remote debugging. I specify the program I want to debug and my User ID so that I am only debugging the program when I initiate it. In the next area of the screen, is where you session type of TCP, port number 8001 and workstation IP address 10.12.123.111.
Now if you run your transaction the Debugger should stop when it reaches your program. Notice the Platform is zOS 390X which indicates that the IBM Debug Tool is being used.
Integrated Debugger using DIRECT directive:

For the Display ID field there are multiple options, DIRECT\ipaddress causes the common debug adaptor to use the Integrated Debugger and go directly to the TCPIP address specified.
If you are debugging a COBOL v4 program then you need to tell the Integrated Debugger where to find the listing. So press PF9 to go to the Options panel, and set the AQE_DBG_V4LIST environment variable to a PDS that contains your listings.

If you are debugging a COBOL v4 program then you need to tell the Integrated Debugger where to find the listing. So press PF9 to go to the Options panel, and set the AQE_DBG_V4LIST environment variable to a PDS that contains your listings.

Return and save your profile, and you should see the Generated TEST string at the bottom of the panel similar to one of the following...
Now if you run your transaction the Debugger should stop when it reaches your program. Notice the Platform is ZOS-PICL which indicates that the Integrated Debugger is being used.

Integrated Debugger DBM directive:
Another option is to specify DBM or DBM.userid causing the Debug Manager (DBGMGR in the Host Config Guide) to use the RSE connection to find the appropriate user. If you want the debug session to appear on your connected RDz UI then you can simply specify DBM. If you want to send the debug session to another user then you can specify DBM.userid (where userid is the ID of the user on whose connected RDz UI you want to receive the debug session).

Debug Tool does provide a JCL to start up a DBGMGR STC (we use RD/z code, so it is the same DBGMAGR) hence the load library in the STEPLIB found first will determine which debugger will be used.

This shows the generated string for the LE options. When you run the transaction the Debugger should stop when it reaches the program. Notice again that the Platform is ZOS-PICL which indicates that the Integrated Debugger is being used.

<table>
<thead>
<tr>
<th>Select type and ID of debug display device</th>
</tr>
</thead>
<tbody>
<tr>
<td>Session Type</td>
</tr>
<tr>
<td>Port Number</td>
</tr>
<tr>
<td>Display Id</td>
</tr>
</tbody>
</table>

Generated String: TEST(ALL,'*','PROMPT','TCP8001','DBM.ROGERN') ENVAR("AOE_DEBUG_V4LIST=ROGERN.IDE.LISTING","AOE_DEBUG_TRACE=3")

Repository String: TEST(ALL,'*','PROMPT','TCP8001','DBM.ROGERN') ENVAR("AOE_DEBUG_V4LIST=ROGERN.IDE.LISTING","AOE_DEBUG_TRACE=3")

Profile Status: Inactive. Press PF5 to Activate.

Troubleshooting

Q1: Why won’t the DTCN transaction startup?

A1: Use ‘CEMT I SYSTEM’ to query status. If the Debugtool is set to “Debug”, then issue ‘CEMT S SYSTEM NODEBUG’ to enable the DTCN transaction.

Q2: While testing CICS remote debug, debug session fails with ‘CEE3200S The system detected a program interruption’

A2: The exception was caused by port 8001 being blocked by the firewall. The probe trace mentions that the connection is refused. A possible work around is to switch from using the DIRECT&10.12.123.111 directive to using DBM.ROGERN which will cause Debug Manager to use the RSE connection to find the appropriate userid and the clients corresponding IP address.

Q3: If the User ID used to log into RDz is TESTABC and the login to CICS is TESTXYZ, how can I trigger the debug session using DTCN?
A3: In theory, you should be able to do what you want by specifying TESTXYZ for the Userid and using the DBM approach. When activating the debugger profile, you enter the DBM directive as before by appending the dot '.' followed RDz (TSO) user ID, e.g. DBM.<userID>

Q4: Receiving message “A breakpoint cannot be installed at address 25b1bf52”

A4: The user must be authorized for one of the two RACF classes (or the equivalent in TSS or ACF2):
Please check CICS job output and system console log for RACF or ACF2 messages, verify SVC are installed correctly.

1. class AQE.AUTHDEBUG.STDPGM - is for most users

2. class AQE.AUTHDEBUG.AUTHPGM - is for users who need to debug authorized program. Debugging authorized programs should be reserved only for those users who have a specific need, as users with this access can bypass system protections.