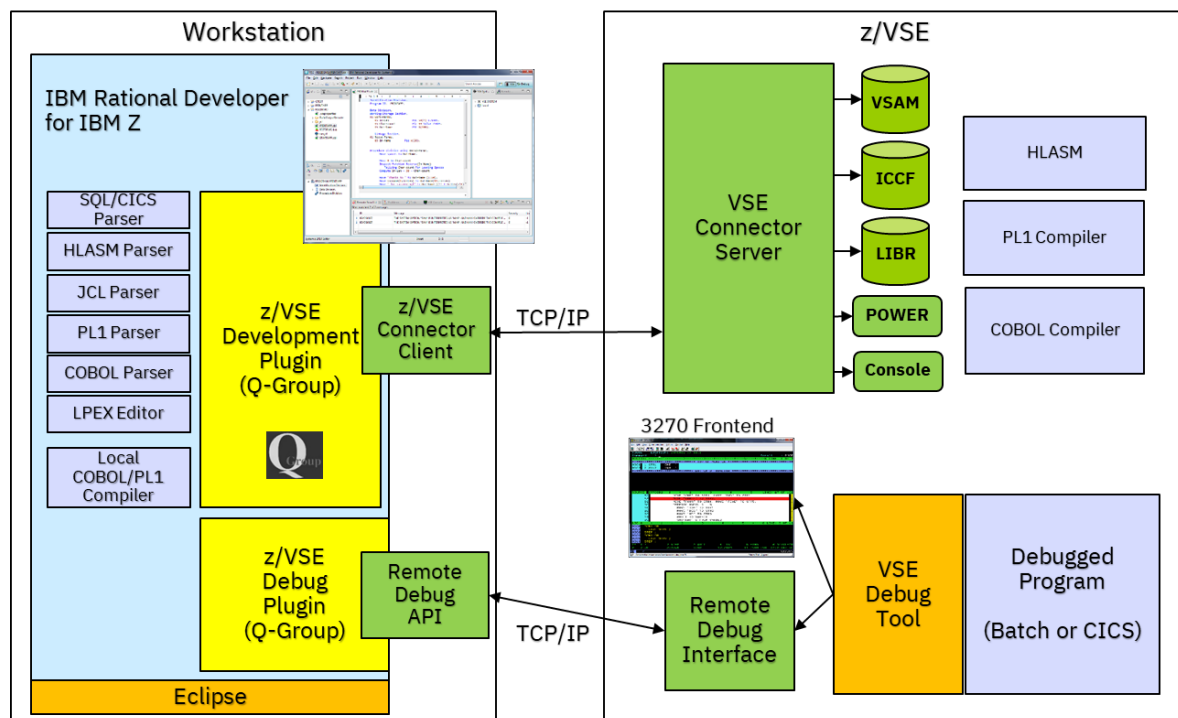


## z/VSE Remote Debug Interface

The z/VSE Remote Debug Interface provides a Java programming interface to remotely debugging an LE/VSE program (batch or CICS).

The z/VSE Remote Debug Interface consists of 2 parts, a host part and a workstation part.

- The **host part** is provided by phase IESRDBGI.PHASE. It hooks in-between the Debug tool back-end and front-end and intercepts the calls to the front-end. In case the remote debug option is selected (via TEST runtime option, details see below), then it connects to the workstation part and passes the requests to it. The host part comes as part of the z/VSE system, starting with z/VSE 6.2.
- The **workstation part** is provided by the JAR file VSERemoteDebug.jar. It provides a Java programming interface to debug z/VSE programs. The workstation part is usually provided as part of the IBM Developer for IBM Z VSE plugin from Q-Group.



**Configuration and activation of the host part:**

The z/VSE Remote Debug Interface requires a **z/VSE V6.2** system as minimum.

Ensure that the **VSE Debug Tool V1.1.1** is installed on your z/VSE system. The default location is PRD2.PROD. Please see here for details:

<https://public.dhe.ibm.com/eserver/zseries/zos/vse/pdf3/manuals/sc268798.pdf>

Please also ensure that you have the most recent PTFs applied for the Debug Tool product, especially PTFs **UK83505** and **UI31207**.

To activate the z/VSE Remote Debug Interface, run the following **activate job**. The activate job is provided in skeleton SKDBGACT in ICCF library 59 and copies the Debug Tool front-end program EQA30TKI.PHASE from PRD2.PROD into PRD2.CONFIG as EQA30ORI.PHASE. It also copies IESRDBGI.PHASE into PRD2.CONFIG as EQA30TKI.PHASE.

**Note:** In case Debug Tool for VSE is installed in a library different to PRD2.PROD, then you must adjust the activate job accordingly.

You can optionally load EQA30TKI.PHASE and EQA30ORI.PHASE into the SVA. If the Debug Tool for VSE phases have been loaded in the SVA, then you must reload these 2 phases after running the activate job.

Activate job (SKDBGACT in ICCF library 59):

```
* $$ JOB JNM=ACTIVATE,CLASS=0,DISP=D
// JOB ACTIVATE - ACTIVATE REMOTE DEBUG INTERFACE
* *****
* THIS JOB ASSUMES THAT THE DEBUG TOOL FOR VSE IS INSTALLED IN
* PRD2.PROD. IF IT IS LOCATED IN A DIFFERENT LIBRARY, THEN YOU
* NEED TO ADJUST THE LIBRARY BELOW.
* *****
* WHEN RUNNING PROGRAMS TO BE DEBUGGED, YOU MUST MAKE SURE
* THAT PRD2.CONFIG IS IN THE LIBDEF PHASE CHAIN IN FRONT OF
* PRD2.PROD (OR WHEREVER DEBUG TOOL FOR VSE IS INSTALLED).
* *****
// PAUSE ACTIVATE THE REMOTE DEBUG TOOL INTERFACE?
// EXEC LIBR,PARM='MSHP'
CONN S=PRD1.BASE PRD2.CONFIG
COPY IESRDBGI.PHASE:EQA30TKI.PHASE REPLACE=YES
CONN S=PRD2.PROD PRD2.CONFIG
COPY EQA30TKI.PHASE:EQA30ORI.PHASE REPLACE=YES
/*
* *****
* DEFINE PROGRAM EQA30ORI TO CICS (GROUP EQA)
* *****
// PAUSE CLOSE DFHCSD FILE IF CICS IS UP : CEMT SE FI(DFHCSD) CLOSE
// LIBDEF *,SEARCH=(PRD2.CONFIG,PRD1.BASE,PRD2.SCEEBASE)
// EXEC DFHCSDUP,SIZE=600K
    DEFINE PROGRAM(EQA30ORI) GROUP(EQA)
        LANGUAGE(ASSEMBLER) DATALOCATION(ANY)
        EXECKEY(USER) RELOAD(NO)
        RESIDENT(NO)
        DESCRIPTION(COPY OF EQA30TKI - DEBUG TOOL FRONTEND)
/*
// PAUSE OPEN DFHCSD FILE IF CICS IS UP : CEMT SE FI(DFHCSD) OPEN
/&
* $$ EOJ
```

When running programs to be debugged, you must make sure that PRD2.CONFIG is in the LIBDEF PHASE chain in front of PRD2.PROD (or wherever Debug Tool for VSE is installed). That way, the Debug Tool back-end loads the Remote Debug Interface as EQA30TKI.PHASE from PRD2.CONFIG, instead of the original phase.

If you plan to debug under CICS, and you have program autoinstall disabled, you must define program EQA30ORI to CICS as follows:

```
PROGRAM:      EQA30ORI
GROUP:        EQA
LANGUAGE:     Assembler
```

This is done by the activate job above in the second step.

You will have to install this program definition afterwards via 'CEDA INTSTALL PROGRAM(EQA30ORI GROUP(EQA))' or restart CICS.

## Debugging z/VSE programs remotely

To debug a z/VSE program, you must supply the **TEST** runtime option. In batch this is usually done via the PARM statement on the EXEC card. For debugging CICS programs you can either include a custom **CEEUOPT** into the to be debugged program, or you enable debugging via the **DTCN** transaction.

For details on the TEST runtime option and how to specify it, please see the 'Debug Tool/VSE V1R1 User's Guide and Reference' manual (SC26-8797):

<https://public.dhe.ibm.com/eserver/zseries/zos/vse/pdf3/manuals/sc268797.pdf>

To use the Remote Debug Interface, you must specify the IP address and port number of the workstation where the counterpart is running. This is done using the session parameter on the TEST runtime option:

```
TEST( , , ,TCPIP&ip-address%port )
```

**Note:** Do NOT specify the preferences file designator (e.g. ':\*'), since this would disable the Remote Debug Interface.

The IP address can be a numeric IPv4 or IPv6 address, or it can be a hostname. The port is separated by a % character. You can omit the port; the default port 8001 used in this case.

Alternatively, you can use the listen-mode. Here the Remote Debug Interface waits for a connection on a specified port number. In the TEST runtime option, you specify the port number as follows:

```
TEST( , , ,TCPIP%port )
```

When specifying the TEST option in batch via the PARM statement, you need to double the & character, because the & character usually denotes a symbolic parameter:

For COBOL:

```
// EXEC program,PARM='/TEST( , , ,TCPIP&&ip-address%port )'
```

For PL1 or C:

```
// EXEC program,PARM='TEST( , , ,TCPIP&&ip-address%port )/'
```

**Note:** Even with the Remote Debug Interface being activated (i.e. copied to PRD2.CONFIG), you can still use the original Debug Tool front-end. The Remote Debug Tool interface checks the session parameter on the TEST runtime option. If it does not find 'TCPIP&...' being specified there, it loads the original Debug tool front-end and routes all calls to it. The original front-end is then loaded as EQA30ORI.PHASE. If you have changed the name to something else, you can specify the name via:

```
// SETPARM DBG$ORI=name
```

**Important:** When running programs to be debugged, you must make sure that PRD2.CONFIG is in the LIBDEF PHASE chain in front of PRD2.PROD (or wherever

Debug Tool for VSE is installed).

In addition, the following statements might be required in the JCL executing the program to be debugged to allow the Remote Debug Interface to use TCP/IP services:

- Ensure the TCP/IP library is in the LIBDEF:  
`// LIBDEF * ,SEARCH=(PRD2.CONFIG,tcpip-lib,PRD2.PROD,...)`
- Point to the desired TCP/IP Stack ID:  
`// OPTION SYSPARM='nn' <-- replace nn with the Stack ID`  
Default is ID 00.
- Specify the EZA interface phase or configure the EZA API Multiplexer.  
`// SEPARM EZA$PHA=phase-name`  
For CSI specify EZASOH99 (this is the default). For BSI specify BSTTIPS1.  
For LFP specify IJBLFPEZ. For using the EZA Multiplexer, specify IJBEZAMX.

Please also see manual "TCP/IP Support" section "Selecting the TCP/IP and SSL Implementation to use":

[https://www.ibm.com/support/knowledgecenter/SSB27H\\_6.2.0/fa2ti\\_select\\_tcpip\\_ssl\\_implementation.html](https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ti_select_tcpip_ssl_implementation.html)

When debugging a CICS program, you need to make sure that above statements are present in the CICS startup job. In addition, the **EZA Task Related User Exit** (TRUE) must be active in CICS. You can activate the EZA TRUE via transaction EZAT manually. However, it is recommended to activate EZA TRUE in the CICS PLTPI (for auto-startup during CICS startup).

Please see manual "TCP/IP Support" section "CICS Considerations for the EZA Interfaces" for details:

[https://www.ibm.com/support/knowledgecenter/SSB27H\\_6.2.0/fa2ti\\_soc\\_prog\\_cics\\_consider\\_eza.html#fa2ti\\_soc\\_prog\\_cics\\_consider\\_eza](https://www.ibm.com/support/knowledgecenter/SSB27H_6.2.0/fa2ti_soc_prog_cics_consider_eza.html#fa2ti_soc_prog_cics_consider_eza)

## ***Problem determination for the Remote Debugging Interface***

In case remote debugging does not work as expected, then please check the following hints and tips to find the reason for the failure:

- 1.) In case the Remote Debugging Interface encounters an error, it usually writes error messages to the console. These messages are prefixed with 'IESC50nn'.
- 2.) Message 'IESC5004E FAILED TO INITIALIZE THE EZA API' indicates that the setup for the EZA interface is incomplete or incorrect. Verify that the LIBDEF statement, the // OPTION SYSPARM='nn' and the // SETPARM EZA\$PHA=. . . statement or the EZA Multiplexer configuration is correct. When debugging under CICS, then also make sure that EZA TRUE is active.
- 3.) Message 'IESC5005E FAILED TO CONNECT TO REMOTE DEBUG SERVER' indicates that the Remote Debugging Interface was not able to connect to the remote system where the debugging server runs. Please check your TCP/IP setup, and verify that the specified IP address or hostname can be reached from your z/VSE system. Often firewalls hinder the connection from being established.
- 4.) If above does not help, then you can turn on the internal trace. You do this by adding the following statement to the job executing the program to be debugged:  

```
// SETPARM DBG$TRC=[SYSLST | SYSLOG | BOTH | OFF]
```

The trace output goes to SYSLST, SYSLOG or both, as specified with DBG\$TRC.  
In addition, you may want to turn on the EZA API trace for the partition that runs the to be debugged program. Enter the following on the console:  

```
EZAAPI TRACE=ON,PART=nn,[SYSLST | SYSLOG]
```

## ***Trademarks***

IBM, the IBM logo, and ibm.com are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at "Copyright and trademark information" at [www.ibm.com/legal/copytrade.shtml](http://www.ibm.com/legal/copytrade.shtml).

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

## ***Comments and Questions***

Comments or questions on this documentation are welcome. Please send your comments to:

[zvse@de.ibm.com](mailto:zvse@de.ibm.com)