

Debug Tool JCL Wizard

Overview

This ISPF edit macro can be used to modify a JCL or procedure member, creating the statements to invoke Debug Tool in various environments. It has the capability to create statements that can be used to invoke Debug Tool for the Terminal Interface Manager, Mainframe Interface, or Remote GUI.

With this tool, you can build control statements to:

- Invoke Debug Tool, accessing the Terminal Interface Manager, Mainframe Interface, or Remote GUI
- Invoke Debug Tool for Language Environment or Non-LE programs
- Include a request to invoke the Automonitor
- Include a request to set AT ENTRY breakpoints
- Include a request to SET WARN OFF or ON
- Define the libraries to search for Debug Tool source information
 - In the event the program name (or CSECT name for assembler) is not the member name of the source file, the wizard will present a list of members for each source file, allowing the user to select the correlated member name.
- Provide a panel to enter the programs which require LDD statements
- Request Code Coverage invocation
 - With or without a Debug session
- Request a Delayed Debug session
- Remove Debug Tool statements
- Show comments depicting how to access subprogram source information prior to being loaded into storage

The location of the statements can be identified by a line command of “A” (after), or “B” (before). If no line command is supplied and more than one program was identified in the JCL or procedure member, the wizard will list all programs, with the capability to select the program you wish to debug. The line command “A” or “B” is required if you wish to provide a procedure step override value.

The Debug Tool JCL Wizard will create in-stream data. Therefore, it will work with a procedure member for JES2, under z/OS 1.13 or later, and with JES3 under z/OS 2.1 or later. Attempting to submit JCL invoking procedures with in-stream control statements will fail if you are not running in one of the environments described above.

Generation of the EQAMDBG DD statement for C/C++ mdbg files is not supported.

Uploading the Debug Tool JCL Wizard files

There are two installation files required.

The first file, *EQA.EXEC.BIN* contains the binary REXX member EQAJCL. The second file, *EQA.ISPPLIB.BIN* contains the binary panel members. Receive these files on the host where you plan to install the Debug Tool JCL Wizard.

Once these files are received, navigate to option 6 of ISPF and issue the command

```
receive inda (EQA.ISPPLIB.BIN)
receive inda (EQA.EXEC.BIN)
```

When receiving the files, provide a name for the PDS libraries that allows READ access by those whom will be using the Debug Tool JCL Wizard.

Installation Instructions

Note: The Debug Tool library hlq.SEQAMOD is assumed to be in the z/OS link list where the batch job is to run. If it is not in the link list, one of the two actions below are required:

- *The library hlq.SEQAMOD may be added to the link list of z/OS LPARs where the EQAJCL ISPF edit macro is expected to be used.*
- *The user will be required to add it to the //STEPLIB or JOBLIB statement of the step or job which is to be debugged.*

This tool consist an ISPF edit macro, and a set of ISPF panels.

It is best to verify if the command EQAJCL is in use. When logged on to the TSO logon procedure used by developers, type the command EQAJCL when editing a member of a PDS. If you see the message “**IKJ56500I COMMAND EQAJCL NOT FOUND**”, then this name is available. Otherwise, choose another name, and rename the EQAJCL REXX exec, or follow the steps in the “Installation by redirecting the code to other libraries” section below.

Option 1: Installation to libraries allocated to the TSO Logon procedure

The member EQAJCL, an ISPF edit macro, should be copied to one of the libraries allocated to the SYSEXEC or SYSPROC libraries which are allocated to the TSO session for all z/OS developers, or by adding the Debug Tool JCL Wizard EXEC library to the TSO logon procedure, as one of the libraries concatenated to SYSEXEC or SYSPROC. The library is an 80 byte FB PDS.

The panels can be copied to an existing ISPPLIB allocated to the TSO session, or the new Debug Tool JCL Wizard ISPPLIB library can be added to the ISPPLIB concatenation in the TSO logon procedure.

Read access to the EXEC and panel libraries should be granted to anyone that wants to use the Debug Tool JCL Wizard.

Option 2: Installation by redirecting the code to other libraries

The ISPF edit macro exec member name is EQAJCL. However, you may wish use a different member name, and install the EQAJCL REXX exec member and panels in PDS or PDSe libraries that are not allocated to SYSEXEC or SYSPROC in the TSO logon procedure.

To redirect the code to another library, install a REXX exec similar to the one shown below. The member name can be DEBUG or another name if you choose. The user would then type DEBUG instead of EQAJCL to invoke the Debug Tool JCL Wizard.

The REXX exec member DEBUG, located in a PDS allocated to SYSEXEC or SYSPROC is shown below, where *hlq* is the high level qualifier for the Debug Tool JCL Wizard libraries.

```
/* This rexx procedure will allocate the required libraries */
/* invoke the EQAJCL REXX routine */
  "ALTLIB ACT APPL(EXEC) DA('hlq.EXEC')"
  "ISPEXEC LIBDEF ISPLIB DATASET ID ('hlq.ISPPLIB')"
  "EX 'hlq.EXEC(EQAJCL)'"
  "ISPEXEC LIBDEF ISPLIB"
  "ALTLIB DEACT APPL(EXEC)"
EXIT
```

Read access to the EXEC and panel libraries should be granted to anyone that wants to use the Debug Tool JCL Wizard.

Debug Tool JCL Wizard Customization:

The REXX exec EQAJCL may require customization, depending on what options are needed.

Delayed Debugging

Delayed debugging is used to allow debugging to initiate at a subprogram. This requires customization as described in the Debug Tool Customization Guide. If this option is enabled, then you will want to update the Inst_DTBASE_SEQAEXEC and Inst_DTWIZARD_ISPLIB fields as described in the comments of the EQAJCL REXX exec.

Code Coverage

Debug Tool Code Coverage measures test case code coverage in application programs that are written in COBOL, PL/I and C and compiled with certain compilers and compiler options. The code coverage libraries xxxx.xxxx.CCPRGSEL and xxxx.xxxx.CCOUTPUT must be defined in the EQAOPTS member residing in

hlq.SEQAMOD, or dynamically created using an EQAOPTS DD statement. The Debug Tool JCL Wizard will automatically add these statements to your JCL, if the variable CODE_COVERAGE_SETUP is set to “YES”. This eliminates the need for the system programmer to change to the EQAOPTS member in *hlq*.SEQAMOD.

If it is set to “YES”, the following statements are generated:

```
//EQAOPTS DD *
EQAXOPT  CCPRGSELECTDSN, '&&USERID.DBGTOOL.CCPRGSEL'
EQAXOPT  CCOUTPUTDSN, '&&USERID.DBGTOOL.CCOUTPUT'
EQAXOPT  CCOUTPUTDSNALLOC, 'MGMTCLAS (STANDARD)                +'
          STORCLAS (DEFAULT) LRECL (255) BLKSIZE (0) RECFM (V,B)  +
          DSORG (PS) SPACE (2,2) CYL'
```

EQAXOPT END

Code coverage requires Enterprise PL/I or Enterprise COBOL using the NOHOOK, or NONE sub parameter of the TEST parameter during compilation.

The Debug Tool JCL Wizard will create Code coverage commands to run without a debug session, or with a 3270 debug session.

Debug Tool JCL Wizard Invocation

The tool can be invoked by simply typing the name of the ISPF edit macro, when editing a JCL or procedure member. For more information, see the Debug Tool JCL Wizard slides.