**Sumário**

[**1.** **Preparando programa para Debug no Changeman.** 2](#_Toc47451115)

[**2.** **Como entrar na ferramenta sob o CICS.** 5](#_Toc47451116)

[**3.** **Preparando arquivo de configuração para a primeira execução.** 7](#_Toc47451117)

[**4.** **Configurando parâmetros para execução no DTZ (Debug Tool)** 9](#_Toc47451118)

[**5.** **Executando um programa com o Debug Tool no CICS.** 10](#_Toc47451119)

[**6.** **Descrição das PF´s.** 15](#_Toc47451120)

[**7.** **Monitorando variáveis** 20](#_Toc47451121)

[**8.** **Lista de comandos e PF´s.** 24](#_Toc47451122)

[**9.** **Executando um programa Assembler com o Debug Tool no CICS.** 27](#_Toc47451123)

[**10.** **Apêndice I – Inspecionando variáveis na memória.** 29](#_Toc47451124)

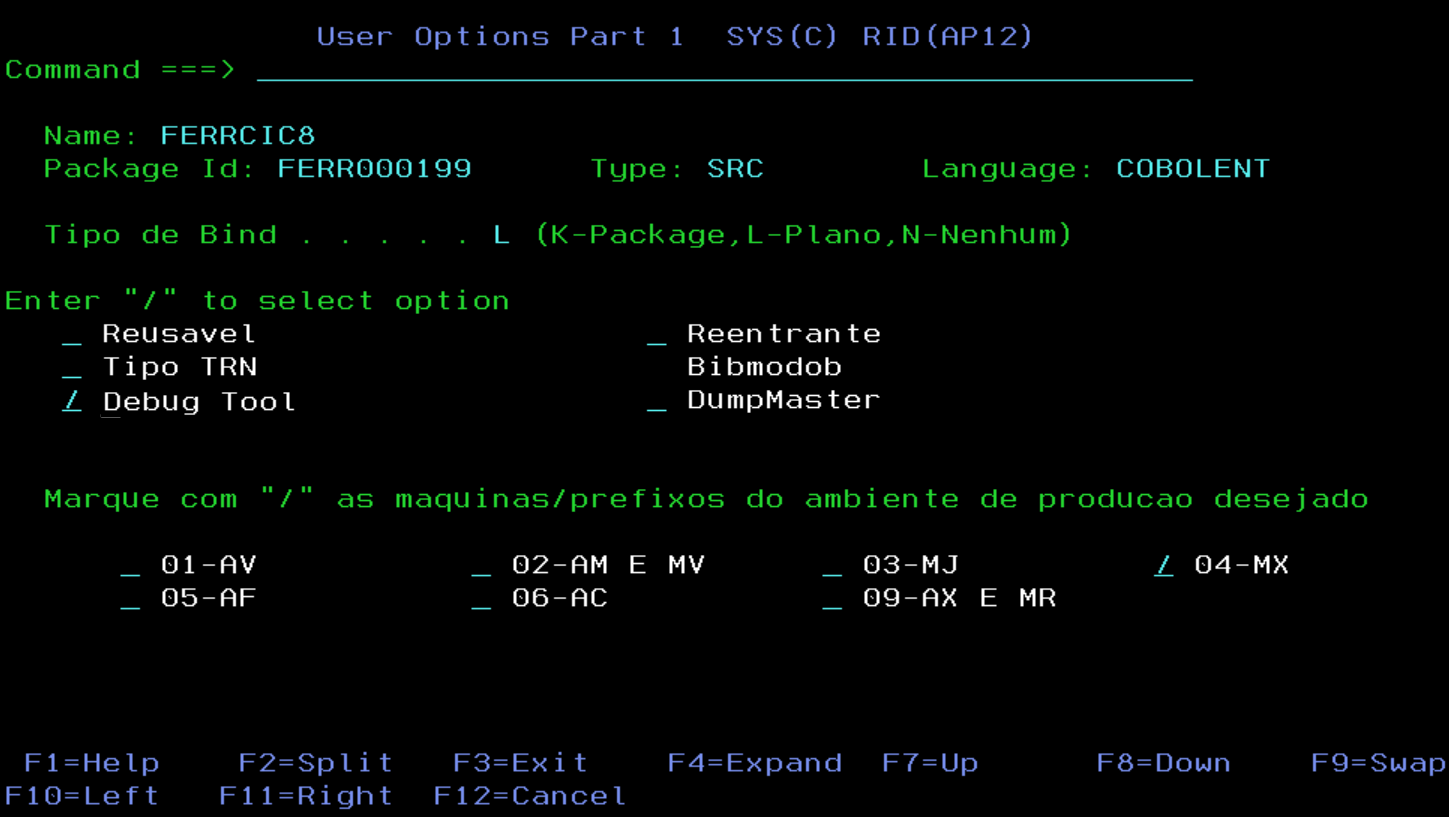
[**11.** **Apêndice – Arquivos de configuração** 31](#_Toc47451125)

[**12.** **Apêndice IV – Link para manuais da IBM.** 32](#_Toc47451126)

**IBM – Debug Tool - CICS**

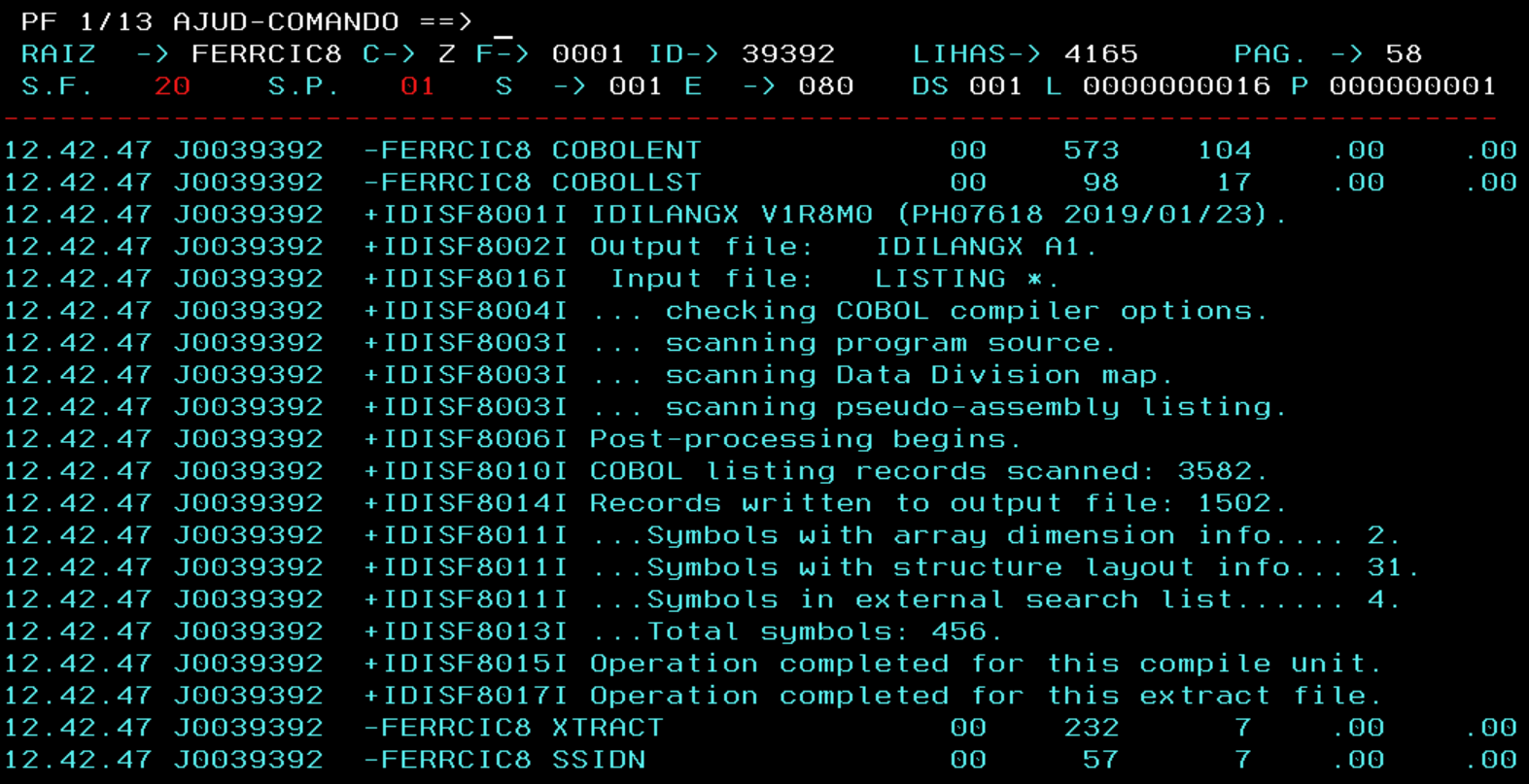
# **Preparando programa para Debug no Changeman.**

O Debug Tool utiliza a listagem de compilação para fazer o link entre os comandos Cobol / Assembler e a instruções de máquina, para isto é necessário marcar com “/” a opção “Debug Tool” (que atualmente ainda está como “Trace Master”) na tela de compilação (Stage) do Changeman.

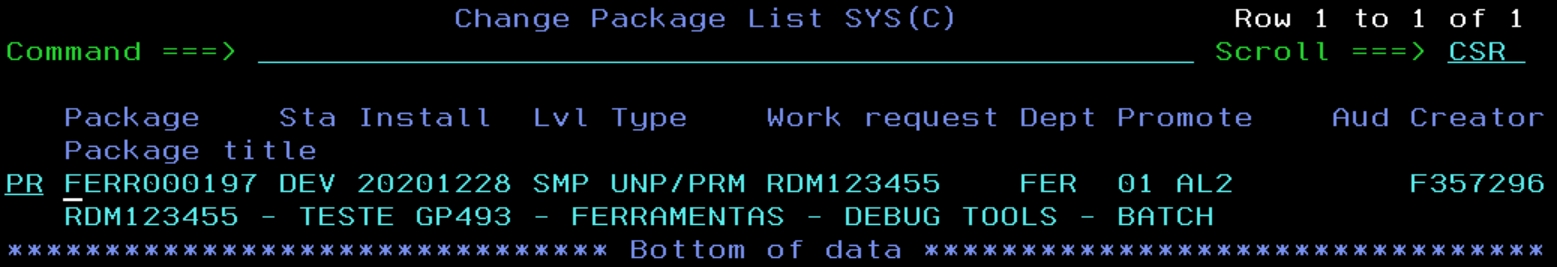


Na execução do JOB de compilação a listagem do programa será direcionada para o arquivo AV.CHGM.DTZ.LANGX.ONL que será utilizada no processo de Debug.

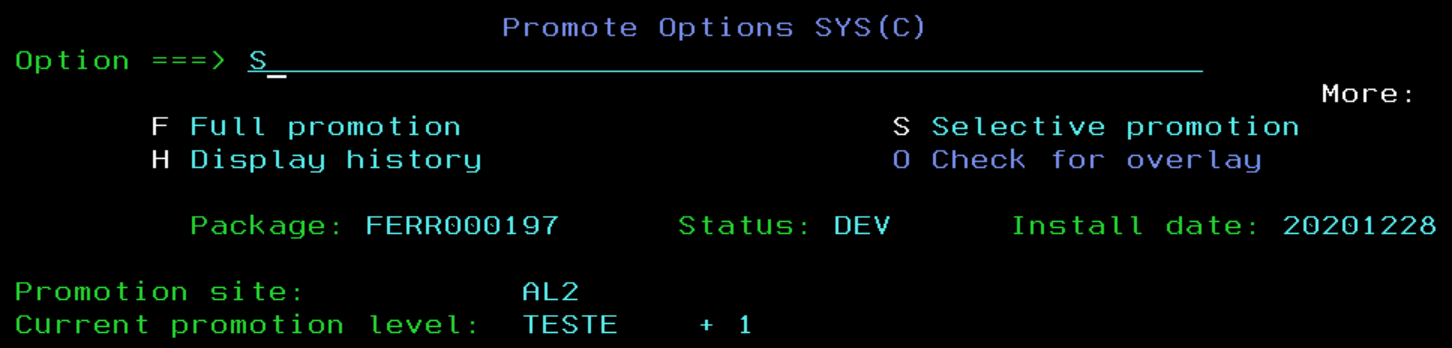
Na SYSOUT do JOB de compilação pode ser encontrada algumas mensagens IDISF80xxI que são geradas pelo Extrator do Debug.

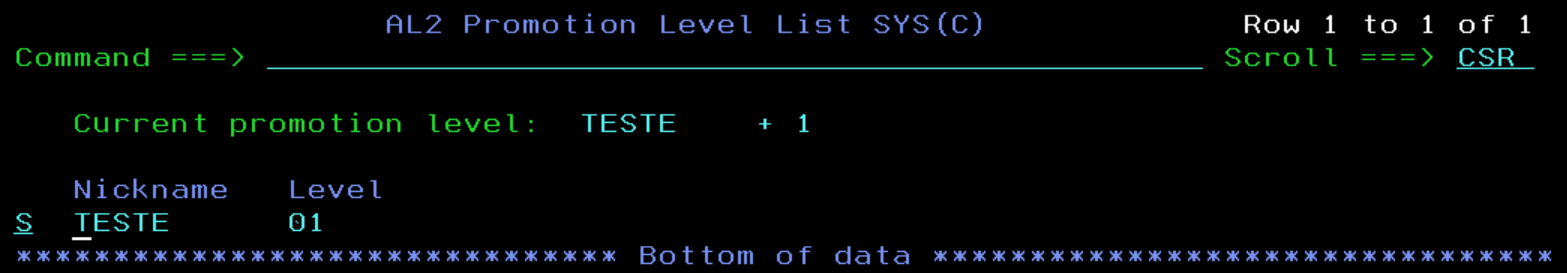


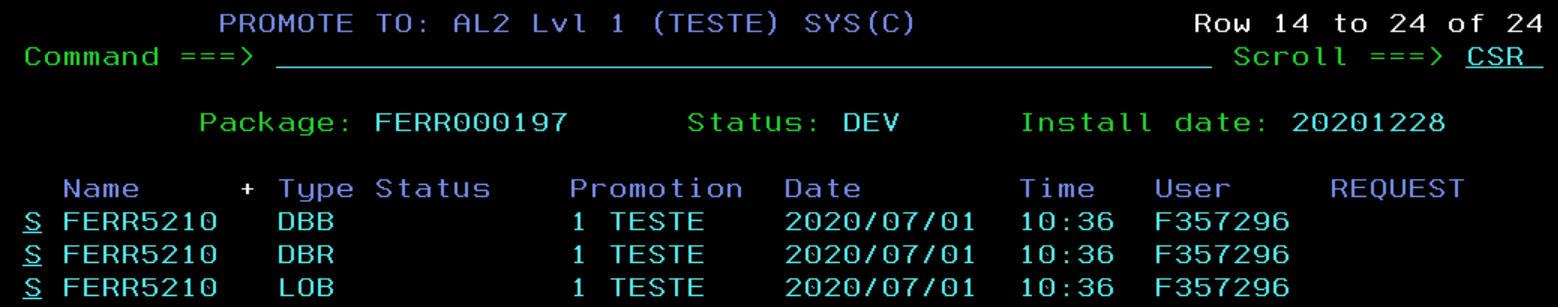
Após a compilação, devemos realizar o processo de **PROMOTE** para o ambiente de TU.











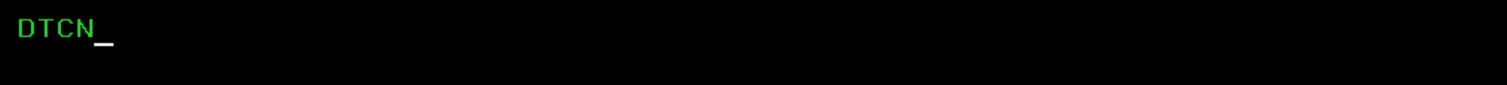
# **Como entrar na ferramenta sob o CICS.**

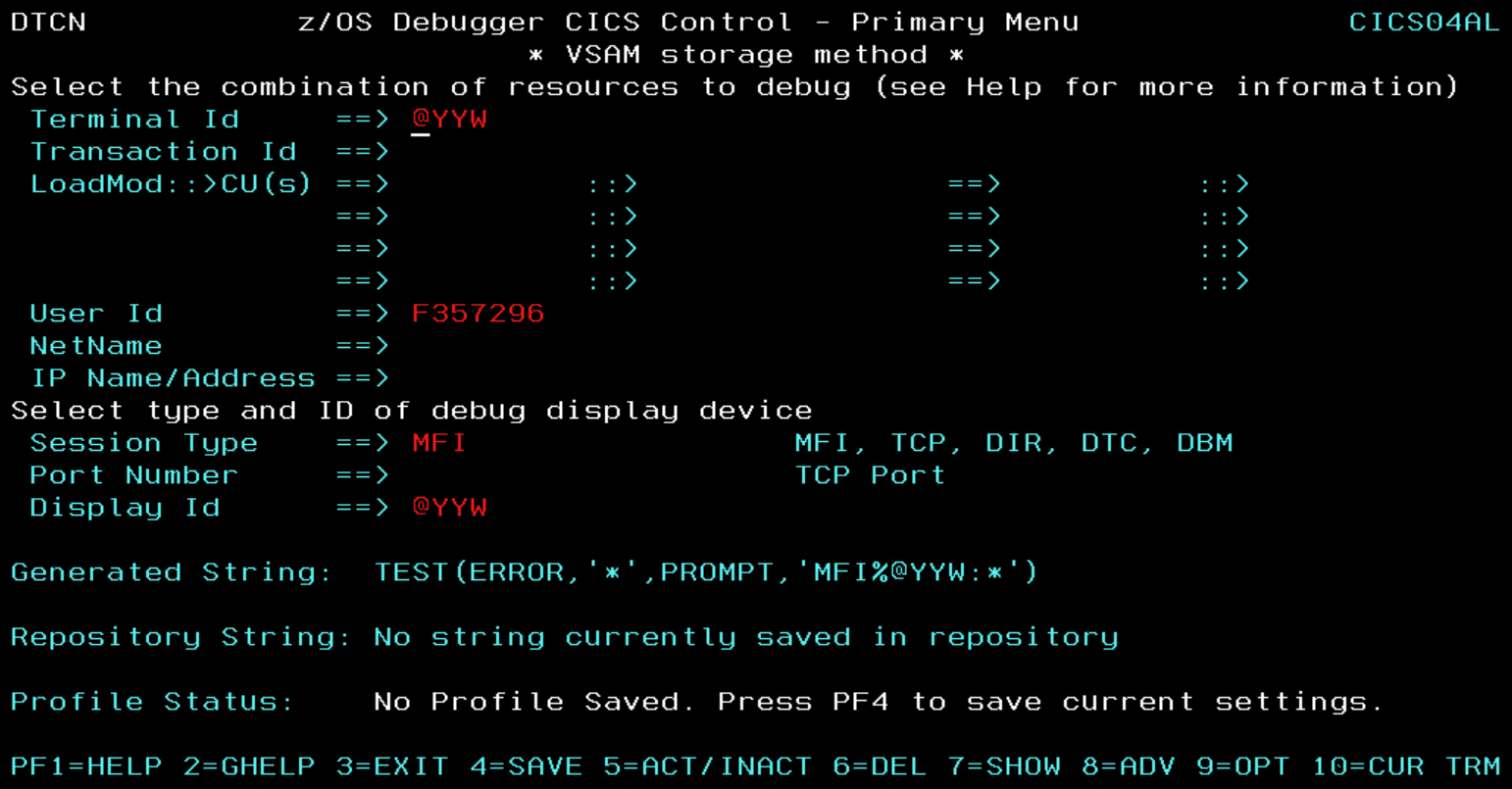
Faça o logon no CICS onde a aplicação irá executar.





Entrar na transação “DTCN” (z/OS Debugger CICS Control), onde serão feitas as interações com a ferramenta.

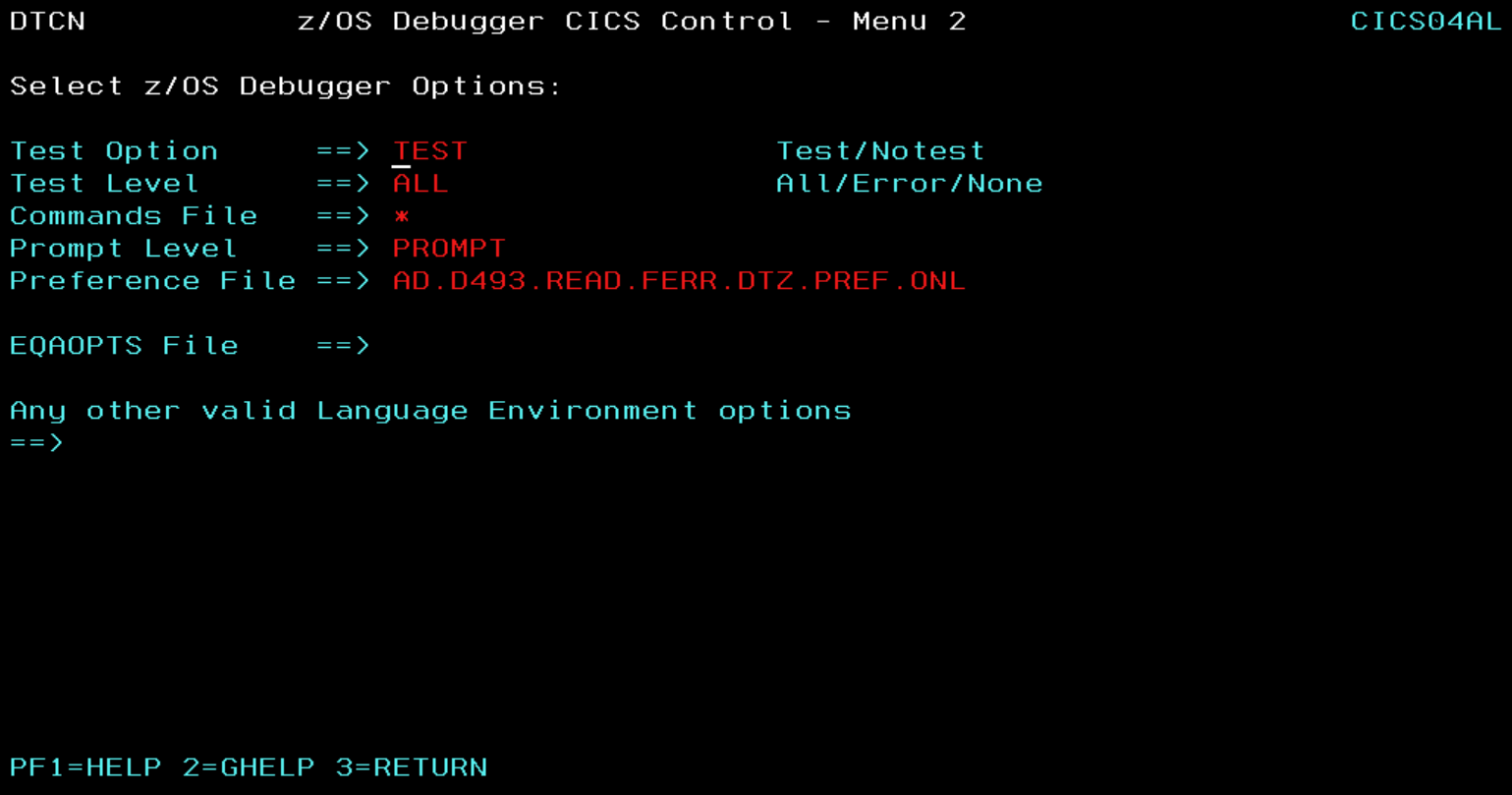




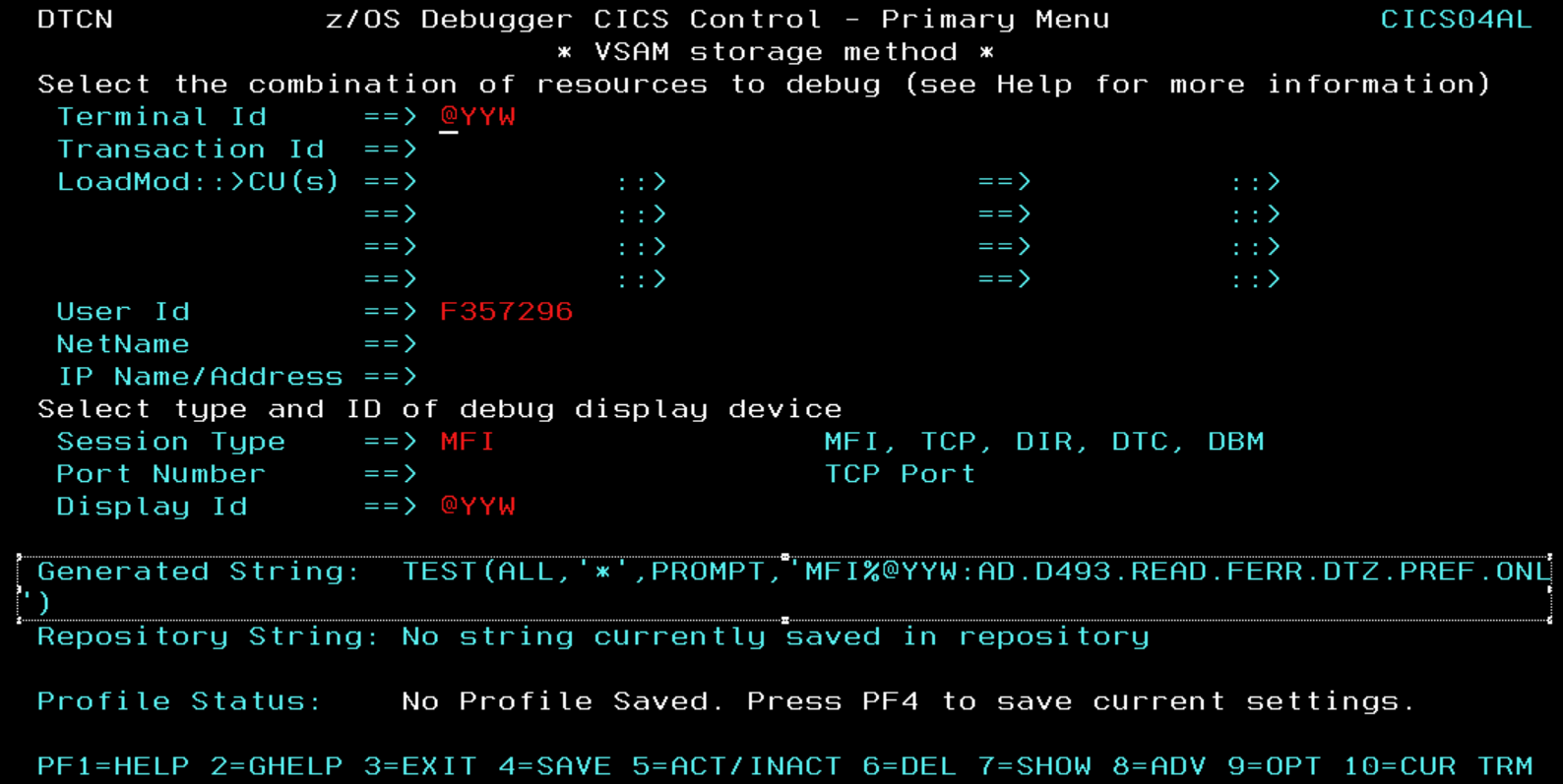
# **Preparando arquivo de configuração para a primeira execução.**

Na tela inicial da transação DTCN, utilize a <Pf9> (“9=OPT”) para entrar na tela de configuração (“Menu 2”), altere o campo “Test Level” para “ALL” e informe o arquivo AD.D493.READ.FERR.DTZ.PREF.ONL (sem aspas) no campo “”, tecle <Enter> para confirmar e <Pf3> para retornar.

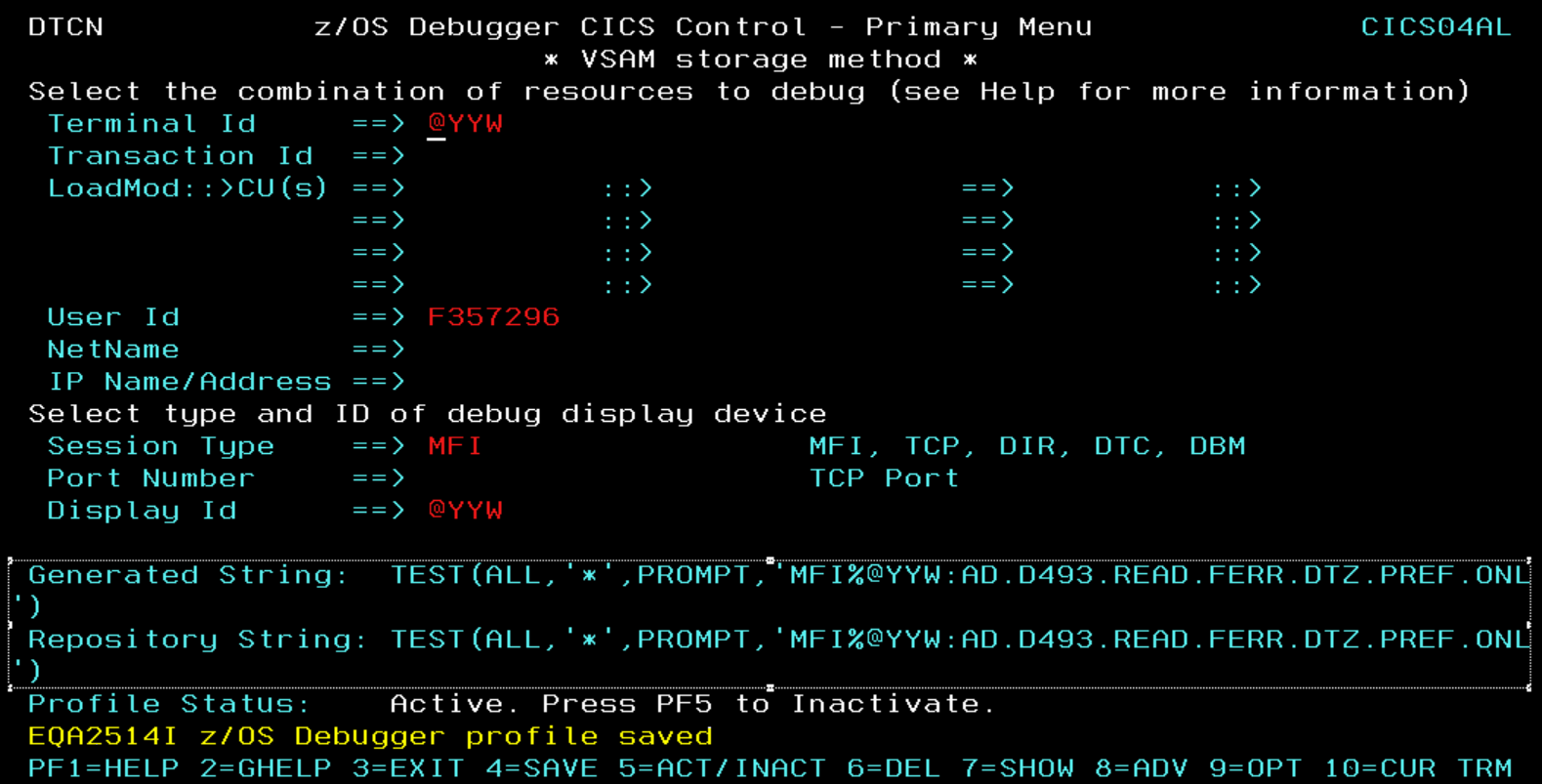
Para TI usar: MP.T78T.PRV.UPDT.DTZ.PREF.ONL



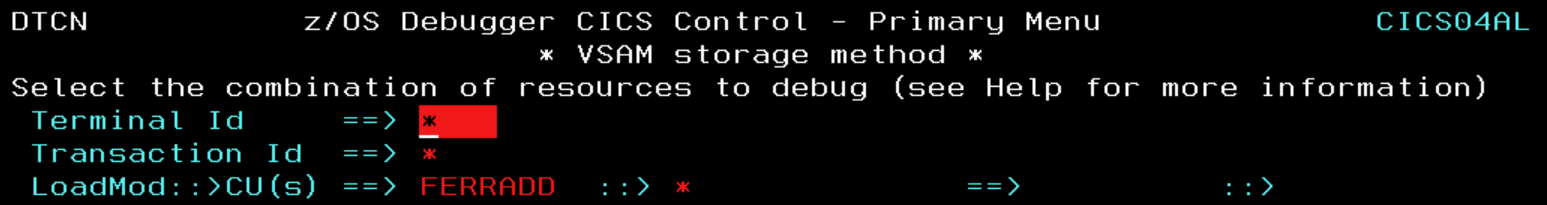
Note que os parâmetros informados aparecem no campo “Generated String”.



Tecle <Pf4> para salvar as alterações.



Caso ocorra a mensagem de “Terminal mismatch”, utilize a <Pf10> (10=CUR TRM) para atualizar com o terminal corrente, isto acontece quando a Profile foi salva anteriormente em outro terminal.

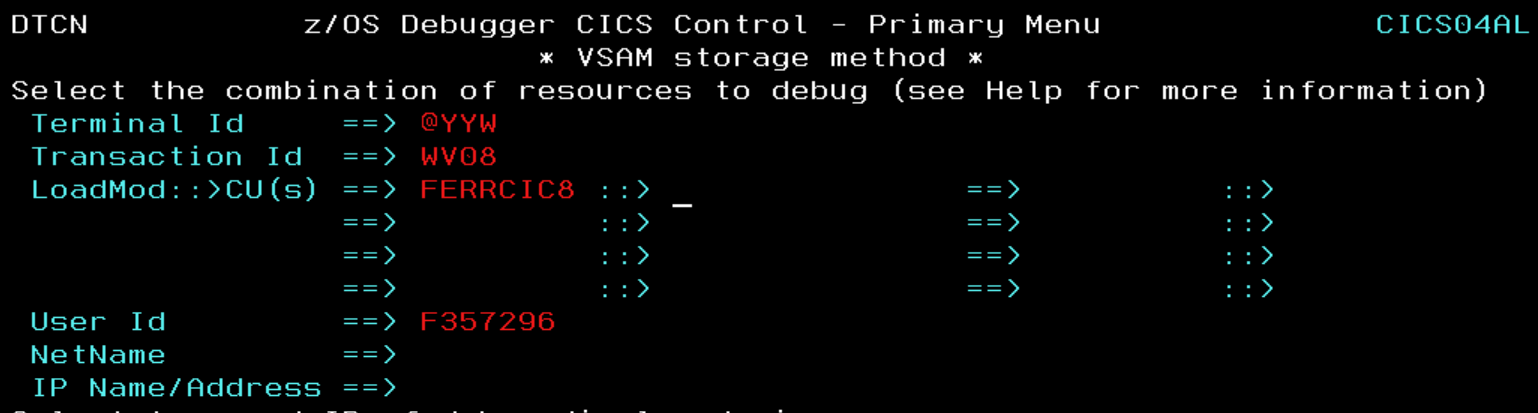




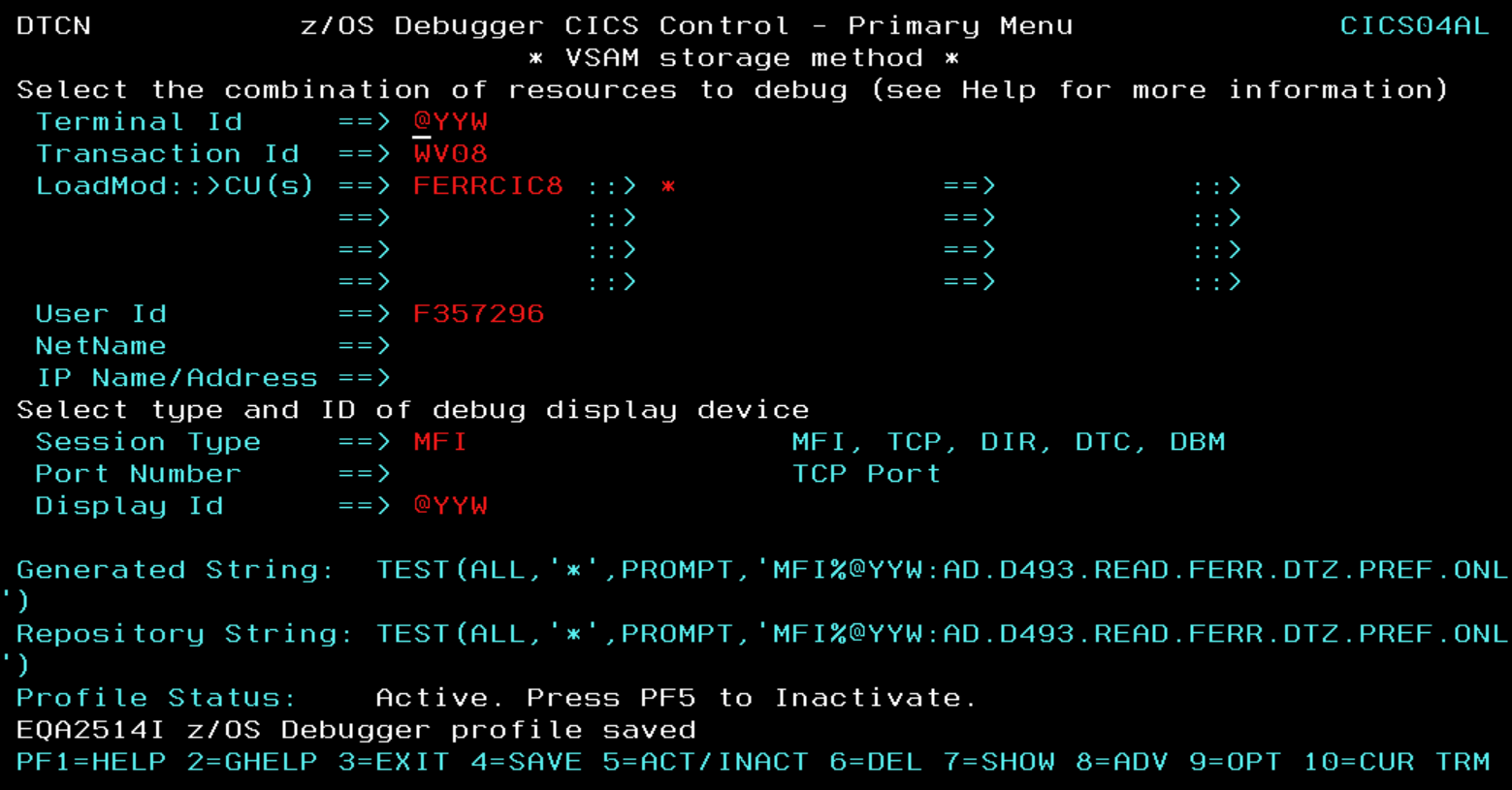


# **Configurando parâmetros para execução no DTZ (Debug Tool)**

Na parte superior da tela do DTCN (z/OS Debugger CICS Control) deverá ser informado a combinação dos recursos que serão “Debugados” (Terminal, Transação, Programa, Usuário), devemos tomar algum cuidado nessas informações, visando restringir a quantidade de transações que serão filtradas para debug.



Tecle <Enter> e em seguida <Pf4> para salvar as alterações.

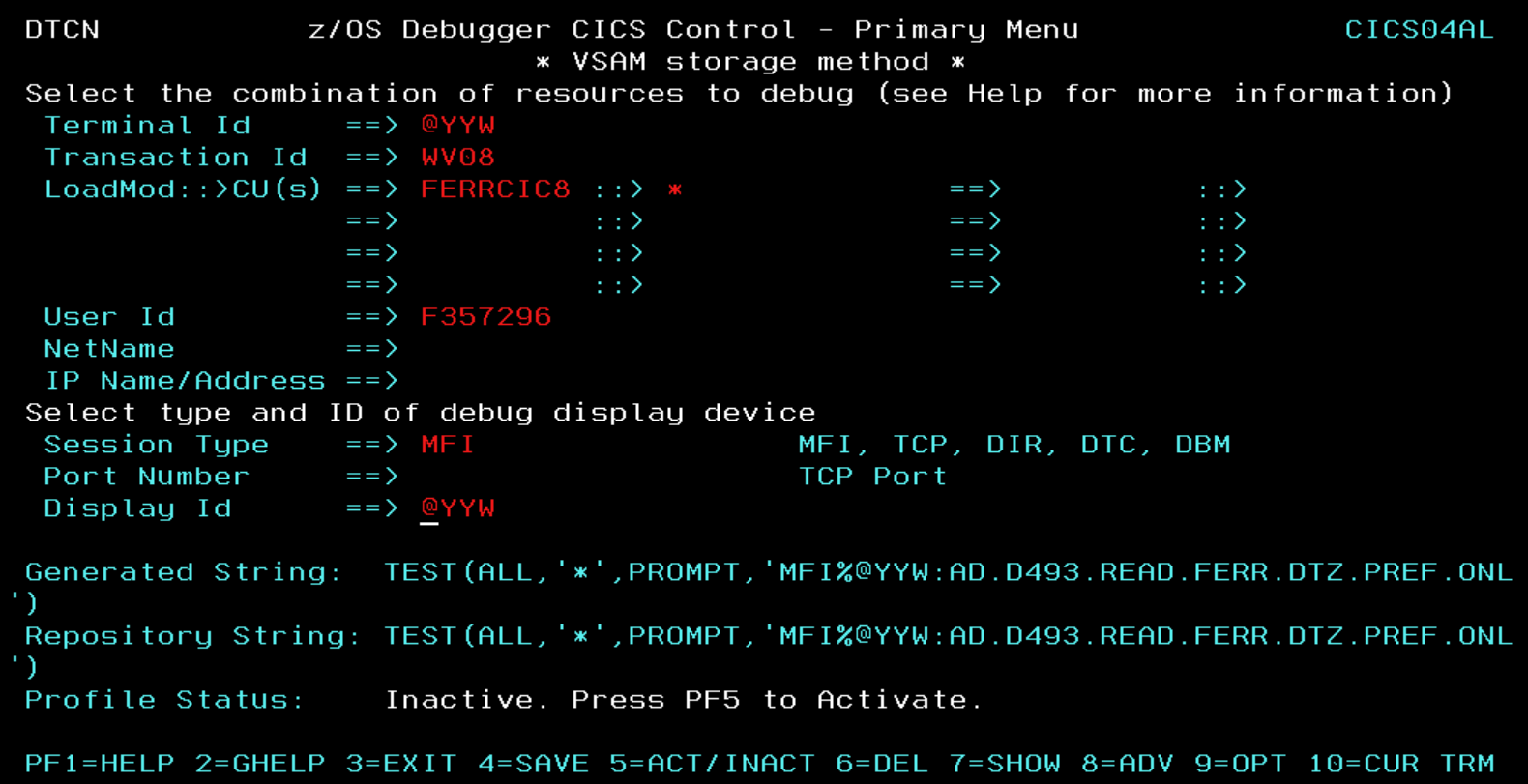


# **Executando um programa com o Debug Tool no CICS.**

Antes de executar o programa, é necessário ativar a “Profile” de captura, isto é feito através da <Pf5> (5=ACT/INACT ), essa PF funciona no esquema de liga/desliga.

Note a mensagem no final da tela (“Profile Status”) indicando que a Profile esta inativa.

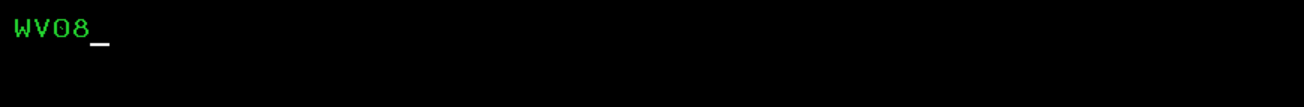




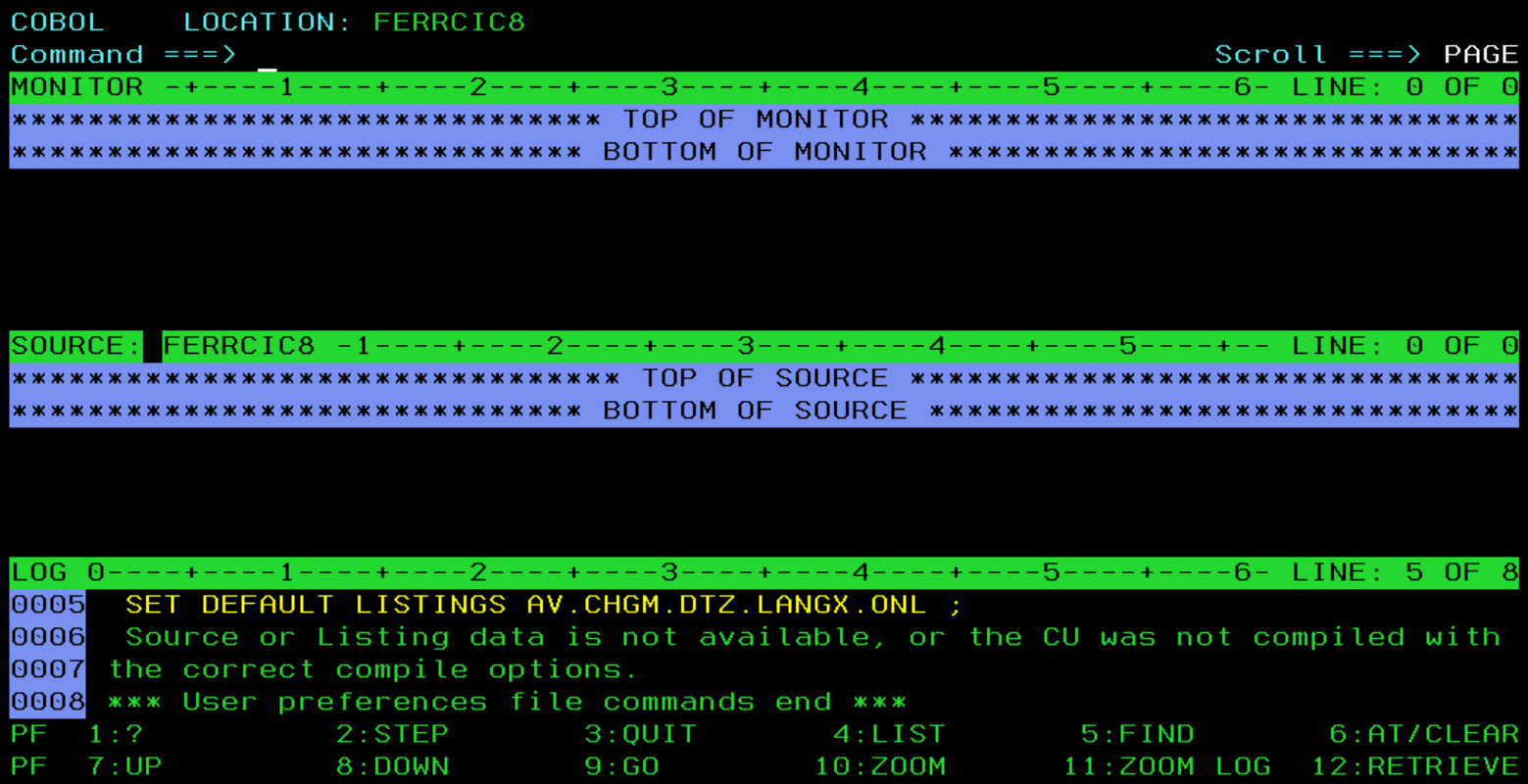
Após a <Pf5>, o Status passa para Ativo.



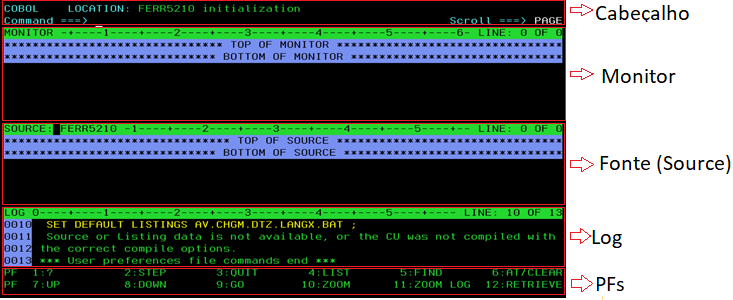
Saia da DTCN com <Pf3> voltando para a tela “limpa” do CICS e ative a sua transação.



Será exibida a tela seguinte:

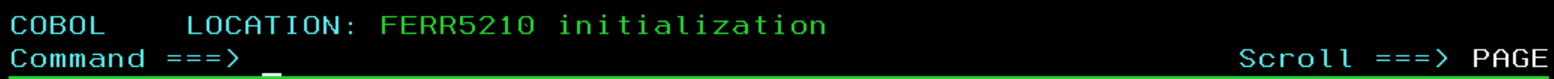


A tela do Debug Tool é dividida em 5 áreas, o Cabeçalho, o rodapé com as PF´s e mais 3 janelas, Monitor, Fonte (Source) e LOG.



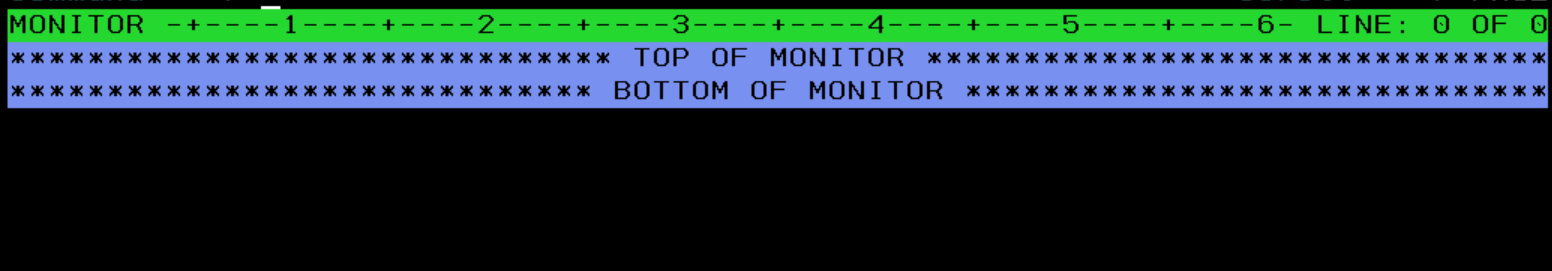
**Área de cabeçalho:**

A primeira linha da área de Cabeçalho mostra o programa e a linha onde o Debug está parado, “esperando” execução, a segunda é a linha de comando, onde serão dadas as instruções para execução.



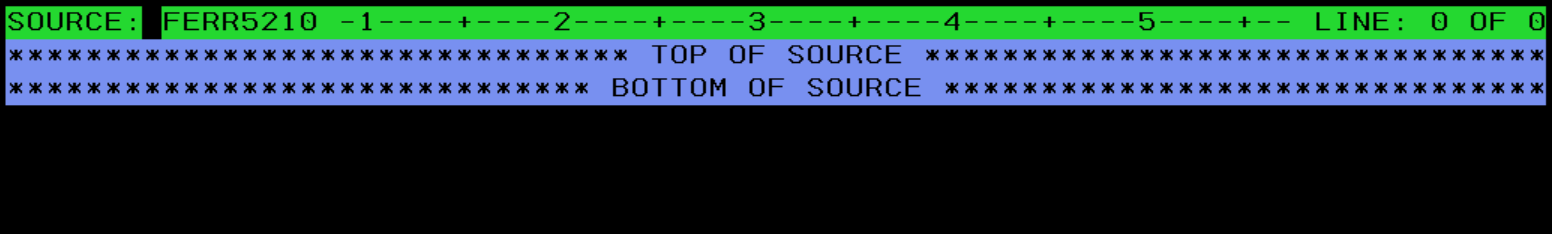
**Janela Monitor:**

Nesta janela serão exibidas as variáveis e seus valores quando os comandos SET AUTOMONITOR ON e MONITOR forem utilizados.



**Janela Source (Fonte):**

Aqui será exibido a listagem do seu programa fonte que está sendo “Debugado”.



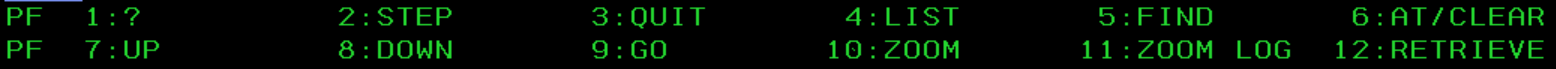
**Janela Log:**

Registra as interações (comandos) do usuário com o Debug Tool e seus resultados e mensagens.



**Área de PF´s:**

Contém algumas teclas de atalho que estão associadas a algumas funções mais utilizadas, facilitando a interação do usuário.

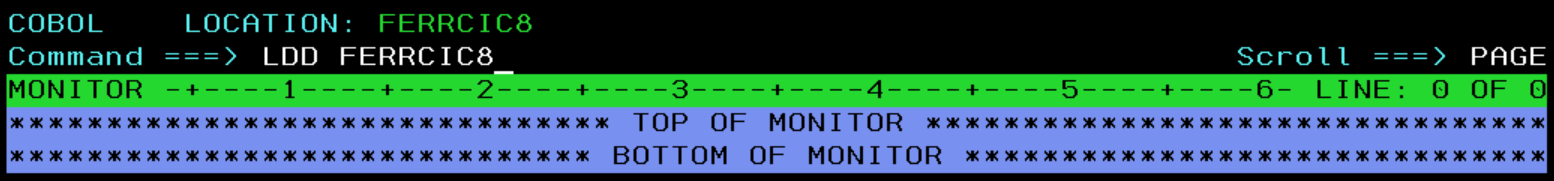


Mais adiante descreveremos as funções da PF´s e alguns outros comandos.

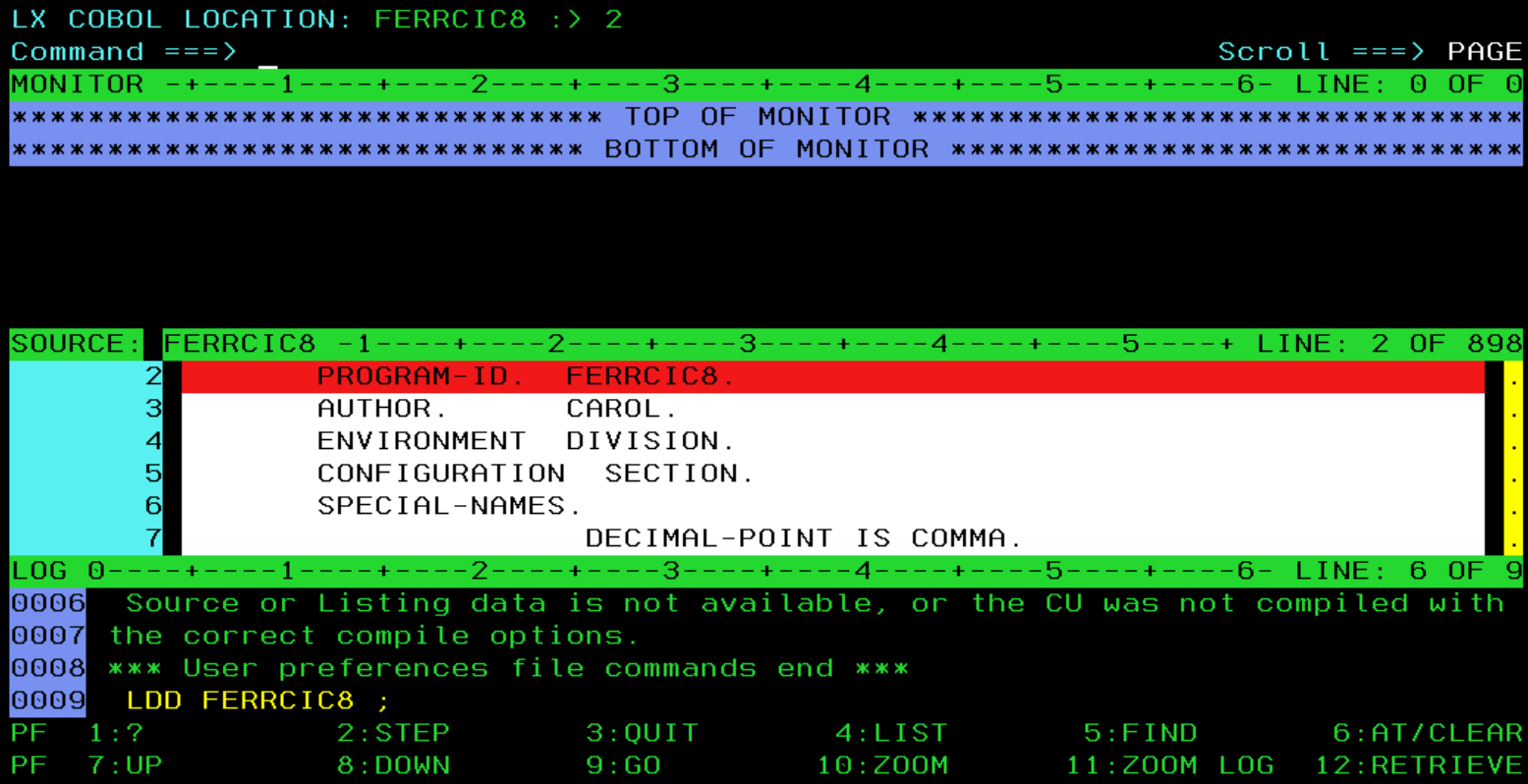
**Carregando a listagem de um programa:**

Para as linguagens Cobol II, MVS Cobol e Enterprice Cobol v3.4 é necessário fazer a carga da listagem do program;a através do comando LDD (Load Debug Data).

Se o programa estiver compilado com Enterprise Cobol v5 (ou superior) a listagem será exibida diretamente sem a necesidade do LDD.



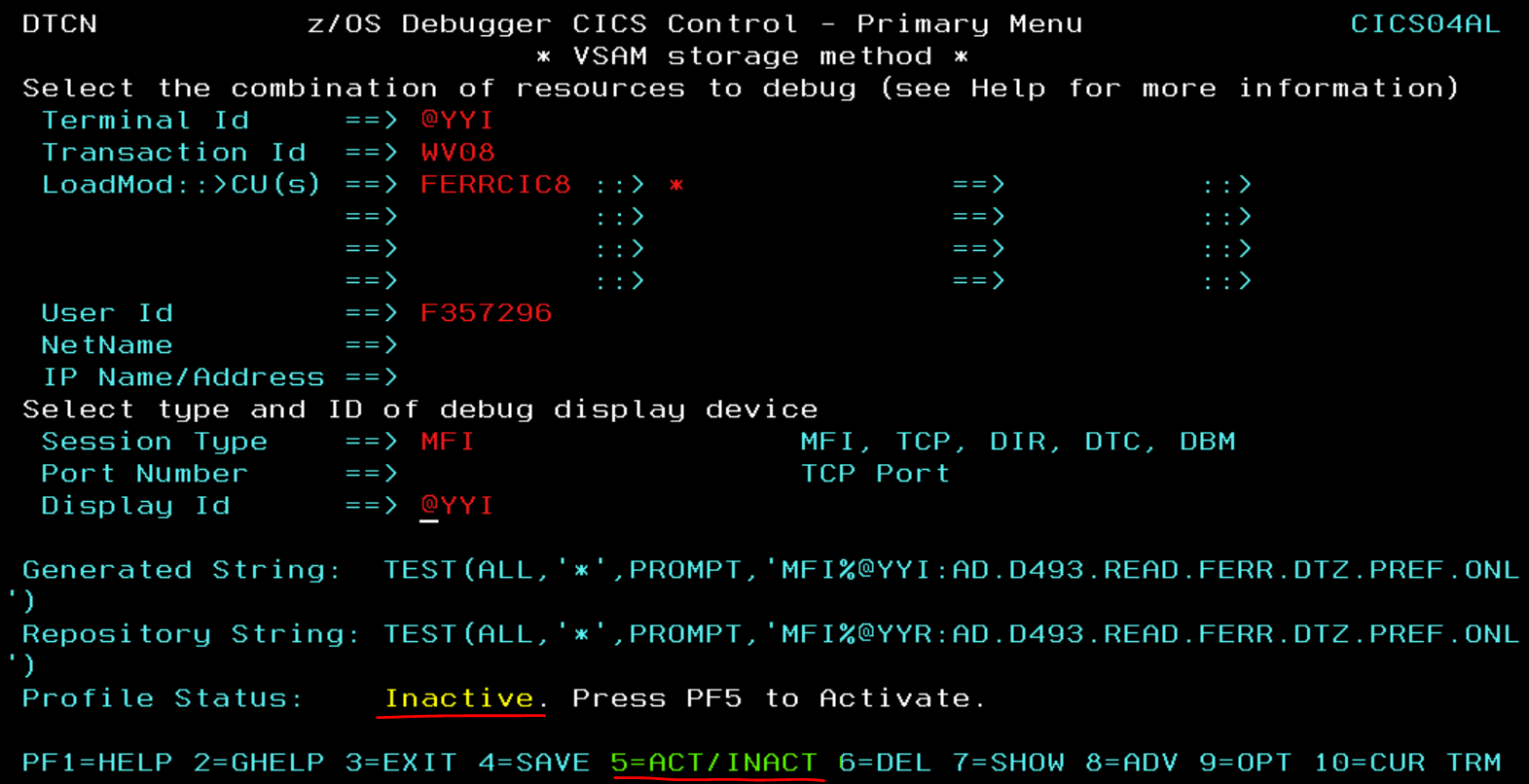
Fonte do programa exibido na janela Source.



Importante: Após encerrar as atividades de Debug aconselhamos (como “boa prática”) desativar a PROFILE utilizada, evitando que algum recurso possa ficar “preso” no CICS.

Entrar novamente na transação DTCN e utilizar a <Pf5> para desativar o profile.



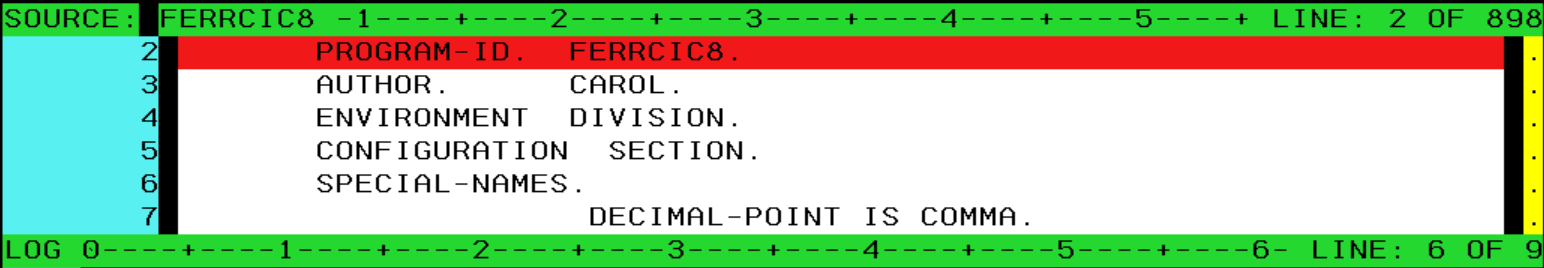


# **Descrição das PF´s.**

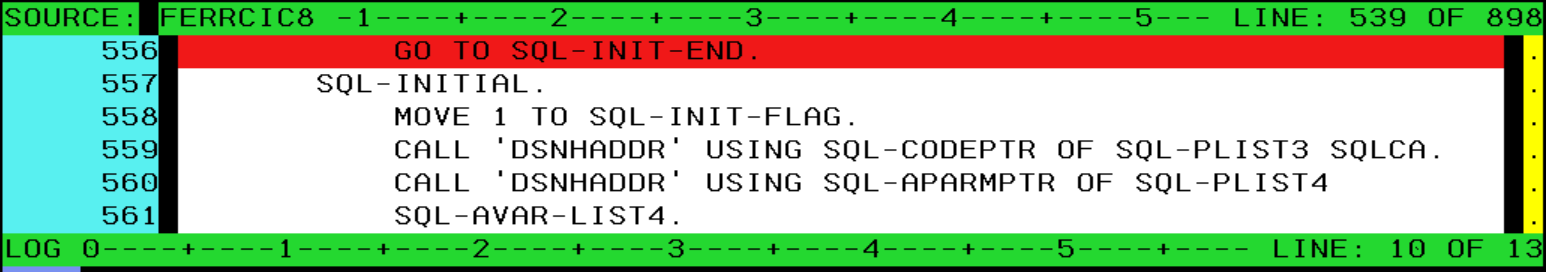
**PF2-STEP:**

Executa imediatamente uma instrução do programa.

Note que quando o programa é carregado a primeira linha do programa fica em destaque em vermelho, são necessários alguns comandos STEP para que o programa seja posicionado no inicio da PROCEDURE DIVISION.

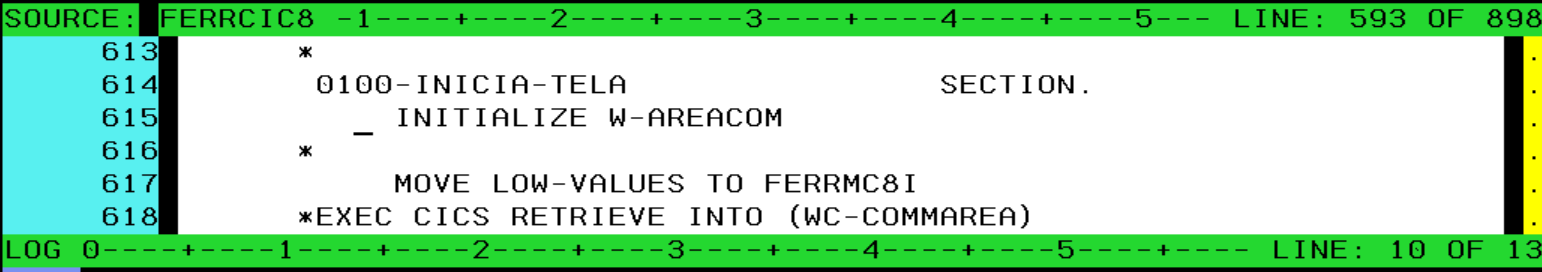


Após acionar <PF2> 3 vezes o programa é posicionado na primeira instrução da PROCEDURE DIVISION.

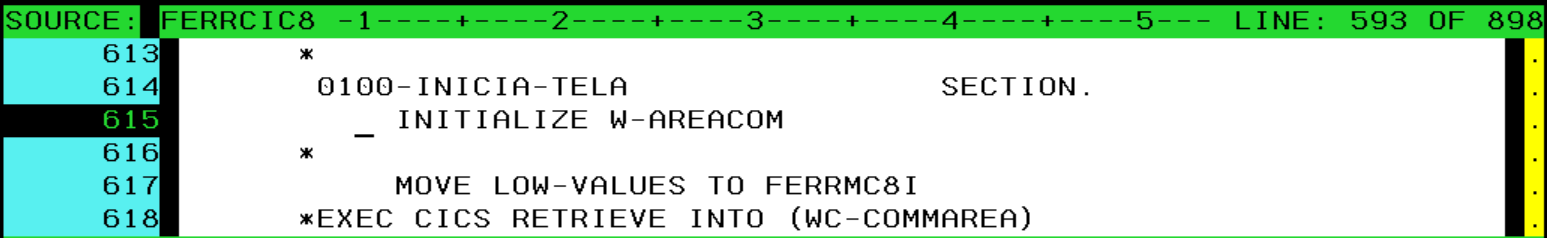


**PF6-AT/CLEAR:**

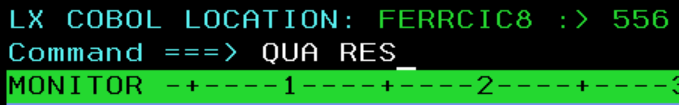
Insere ou retira um Break Point na linha onde o cursor estiver posicionado.

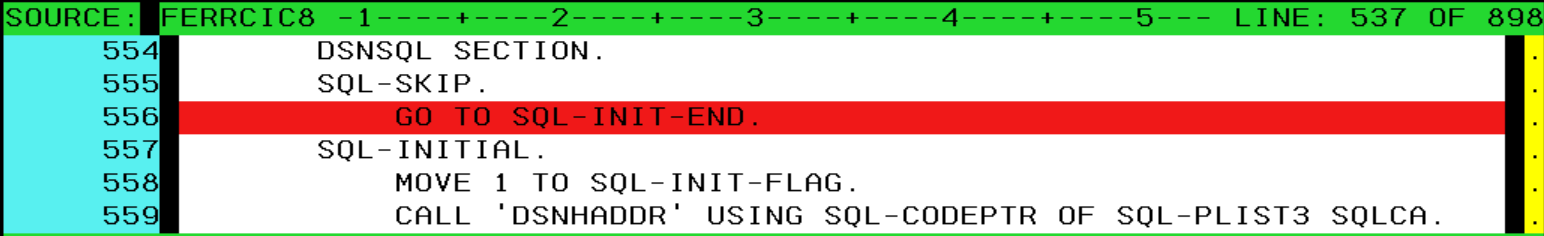


Após <PF6>, note que a linha 615 onde o curso estava ficou em “destaque” indicando um BREAK POINT.



Para retornar ao inicio da PROCEDURE DIVISION, podemos utilizar várias <PF8> ou o comando “QUALIFY RESET” (ou “QUA RES”).

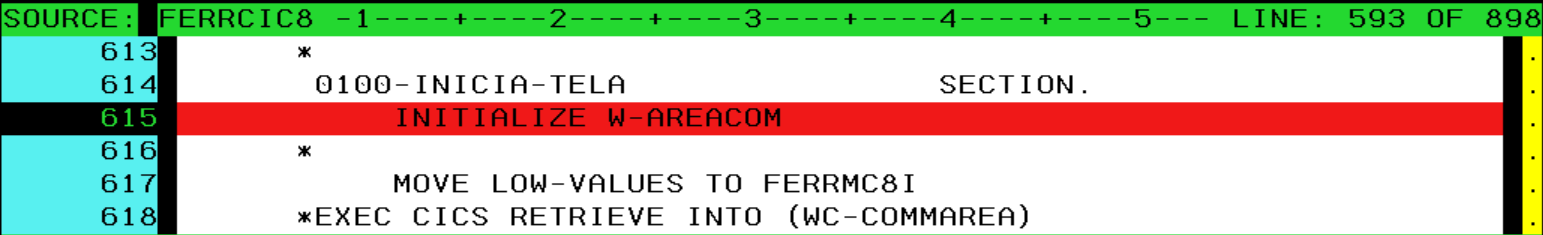




**PF9-GO:**

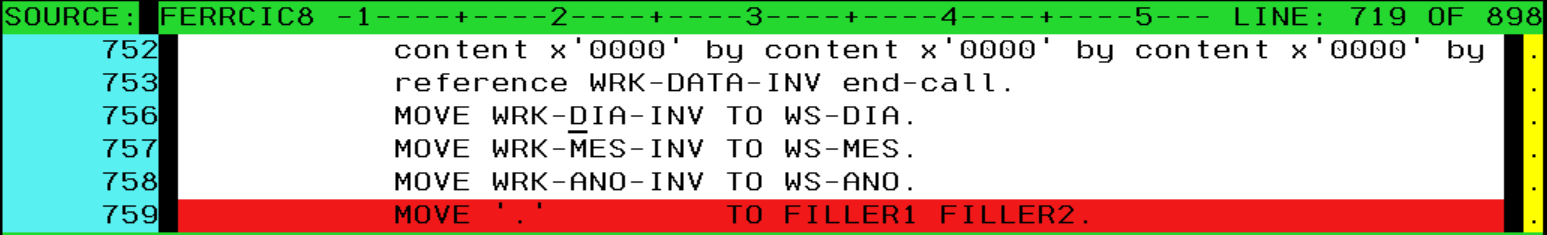
Executa o programa até o próximo BREAK POINT ou até o final caso não encontre BREAK POINT.

Após a <PF9> o programa executou até o BREAK POINT da linha 615.

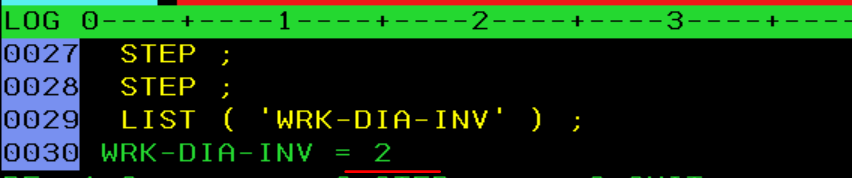


**PF4-LIST:**

Exibe o conteúdo de uma variável onde o cursor estiver posicionado.



Após teclar o “<PF4> - List” o conteúdo da variável é exibido na janela de Log.



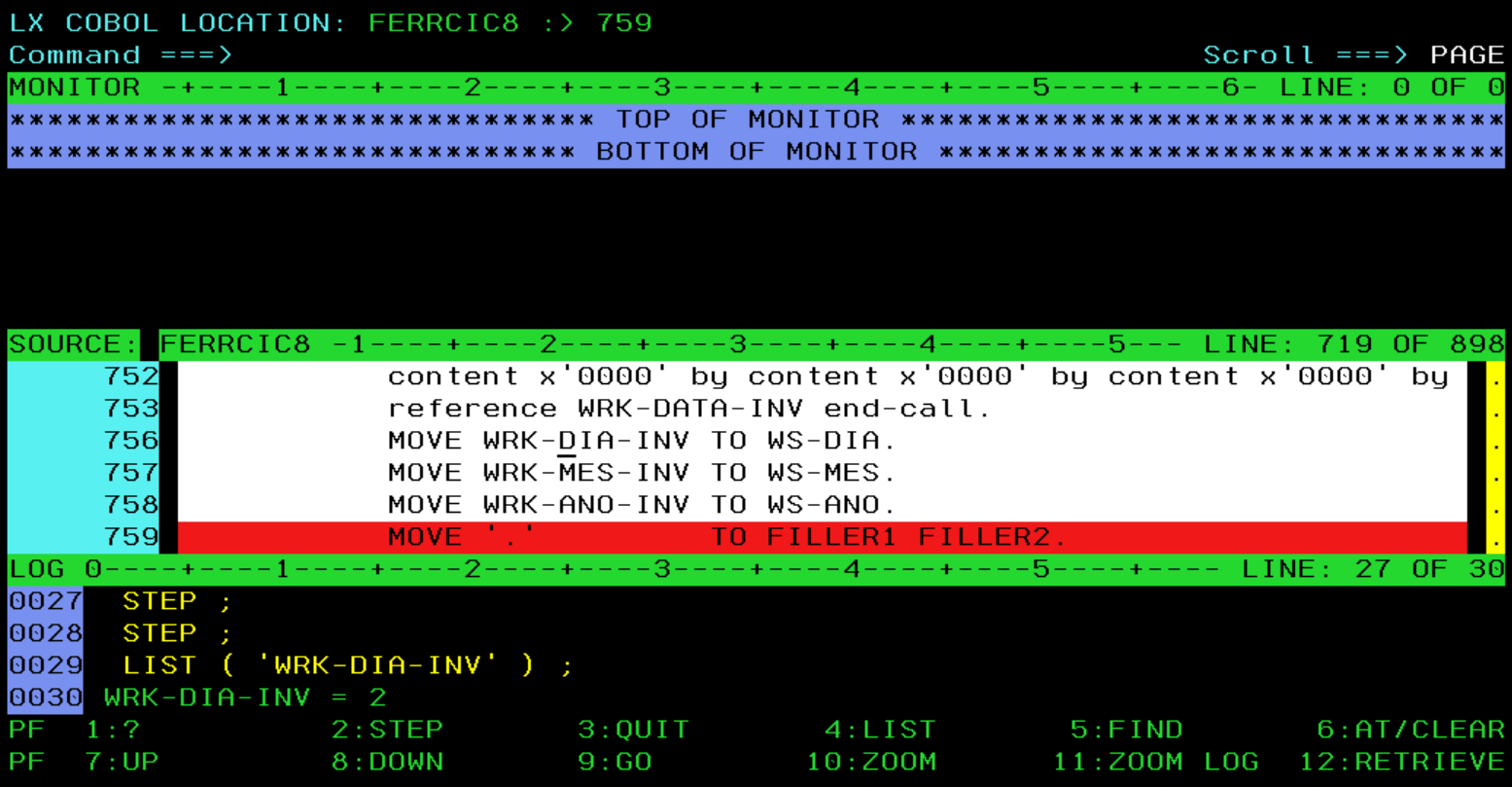
**PF11-ZOOM LOG:**

Expande a janela de Log possibilitando uma melhor visualização, funciona no esquema de liga/desliga, expande ou retrai a janela de acordo com sua situação.

<PF11>, expande a janela

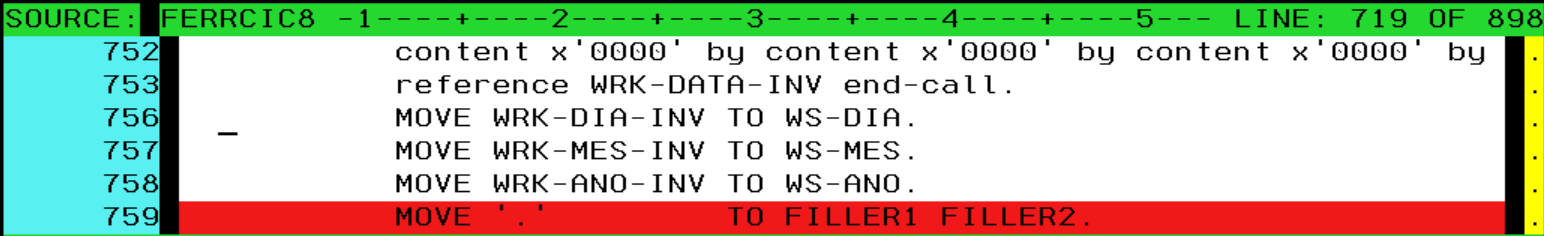


Outra <PF11> volta a janela ao normal.

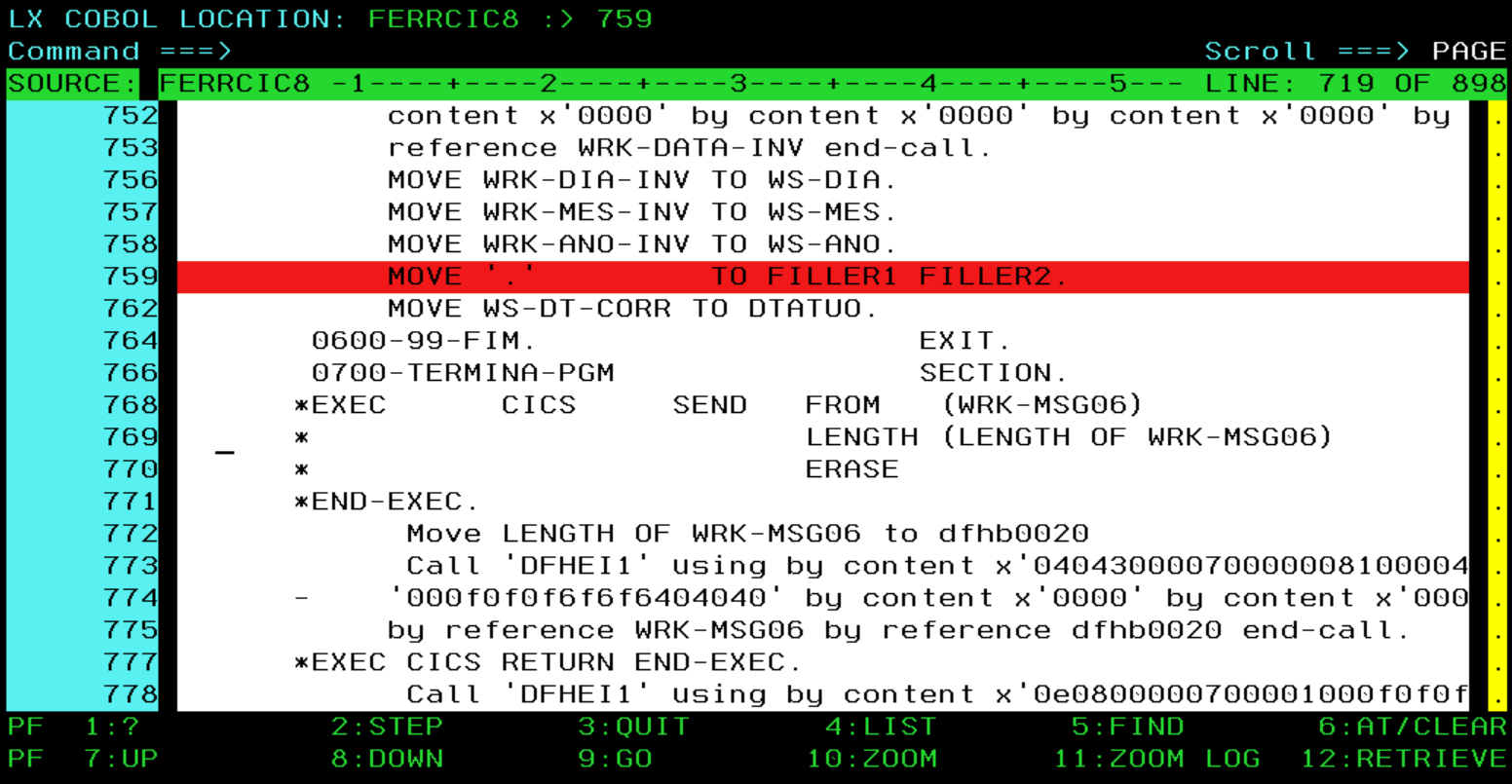


**PF10-ZOOM:**

Expande a janela onde o cursor estiver posicionado possibilitando uma melhor visualização, funciona no esquema de liga/desliga, expande ou retrai a janela de acordo com sua situação.

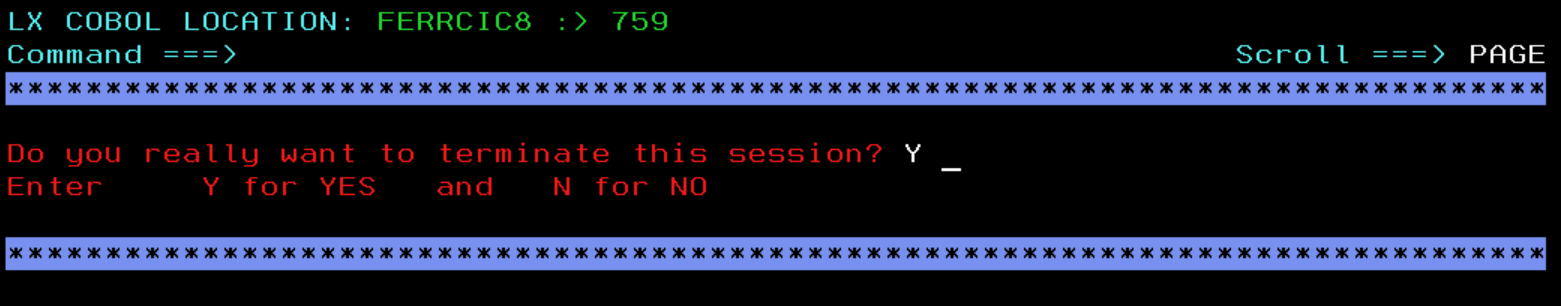


<PF10>, expande a janela

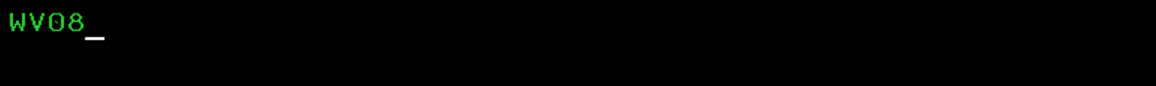


**PF3-QUIT:**

Encerra a sessão de Debug.



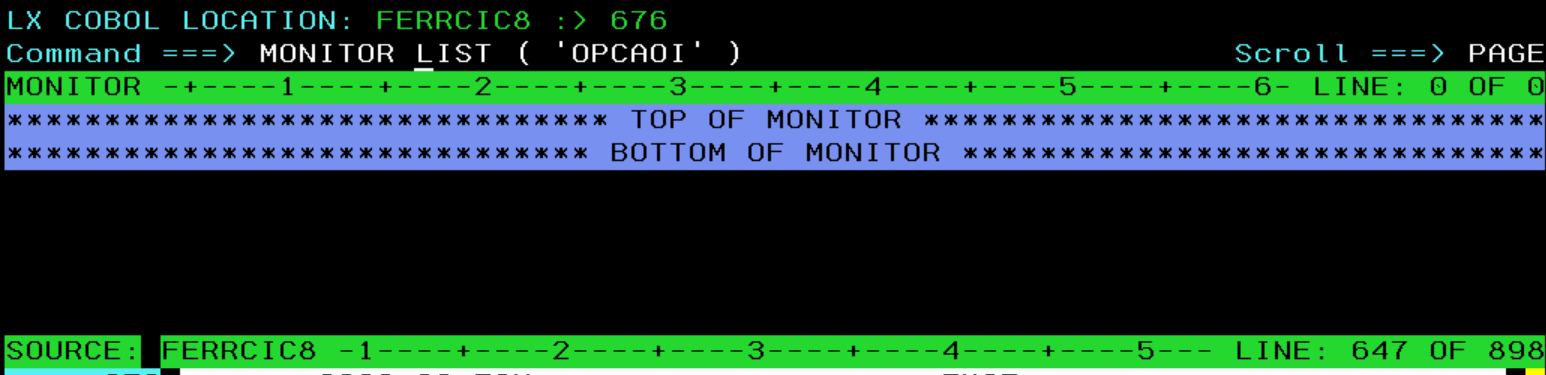
Retorna para a tela inicial do CICS.



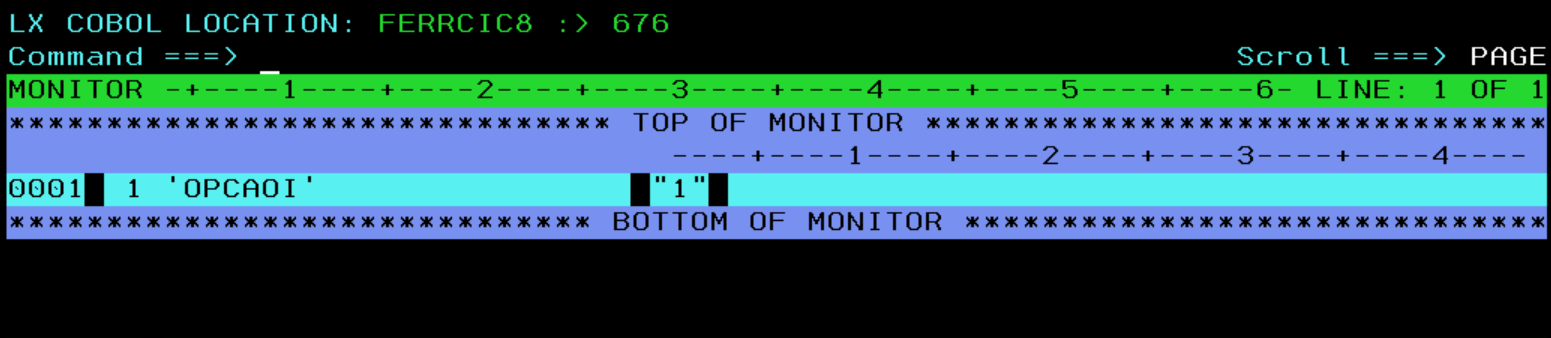
# **Monitorando variáveis**

Modo I: Monitorando variáveis individualmente

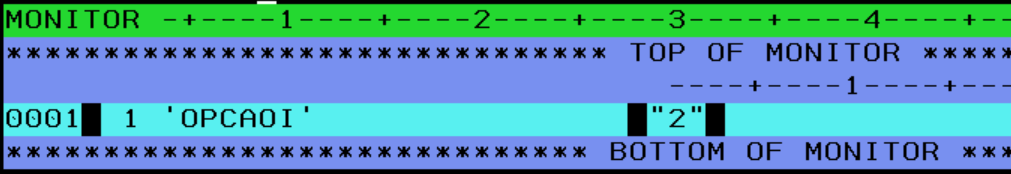
Entre com o comando MONITOR LIST ('nome-da-var')



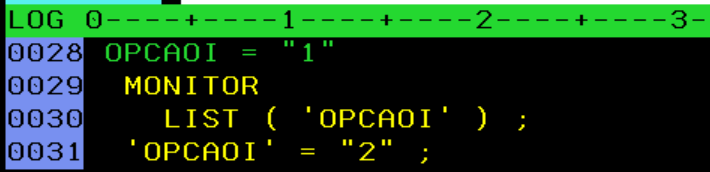
Será exibido o conteúdo da variável na janela Monitor, esta janela ficará fixa na tela sendo atualizada quando a variável sofrer atualização.



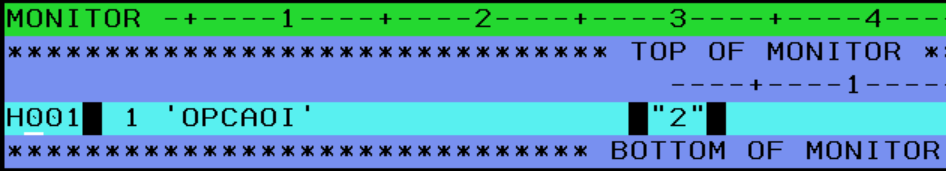
Também podemos alterar o valor da variável, basta digitar o novo valor na janela e teclar <Enter> para efetivar.



Note na janela de Log, que após o <Enter>, foi emitido o comando de atualização do campo.



Exibindo o valor em Hexadecimal, digite “H” no número em frente ao nome da variável e <Enter>.





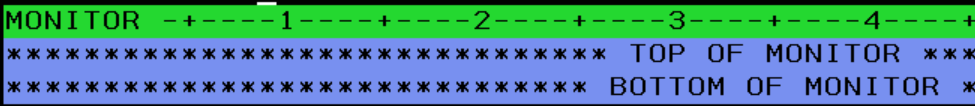
Para voltar ao modo “Default”, digite “D” no número em frente ao nome da variável e <Enter>.



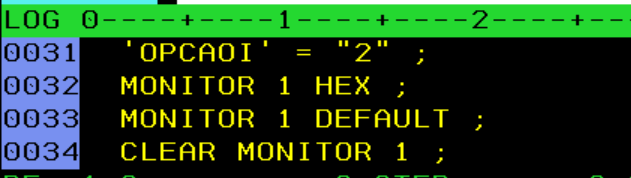


Digite “C” para fechar (“limpar”) o monitor de uma variável.





Note que na janela de Log aparecem os comandos correspondentes:



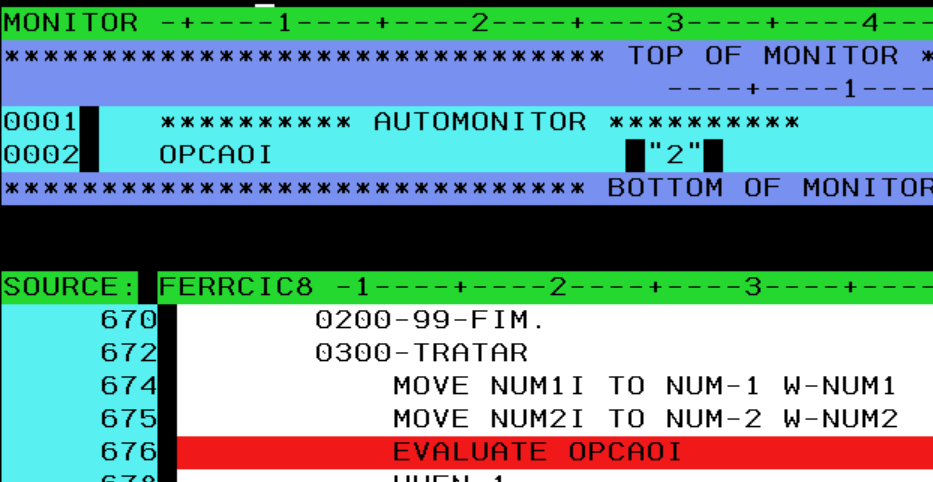
Modo II: Monitorando variáveis automaticamente

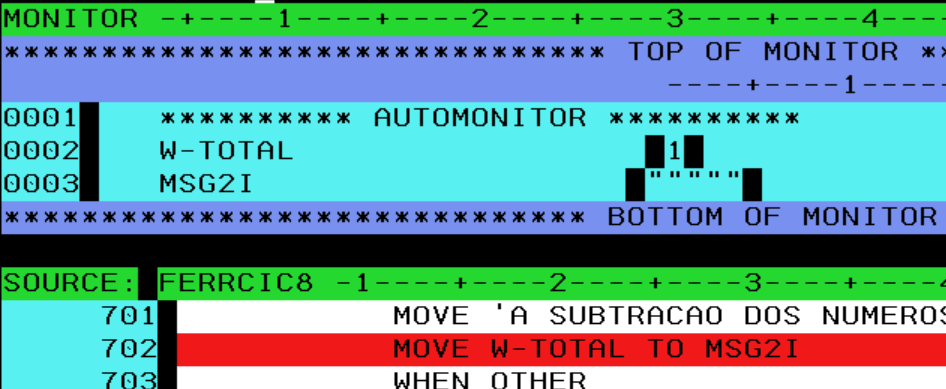
Neste caso, serão monitoradas automaticamente as variáveis envolvidas no comando executado.

Entre com o comando SET AUTO ON <Enter>.



De acordo com as variáveis envolvidas no comando a janela de monitor vai mudando, mostra a variável antes da execução do comando.



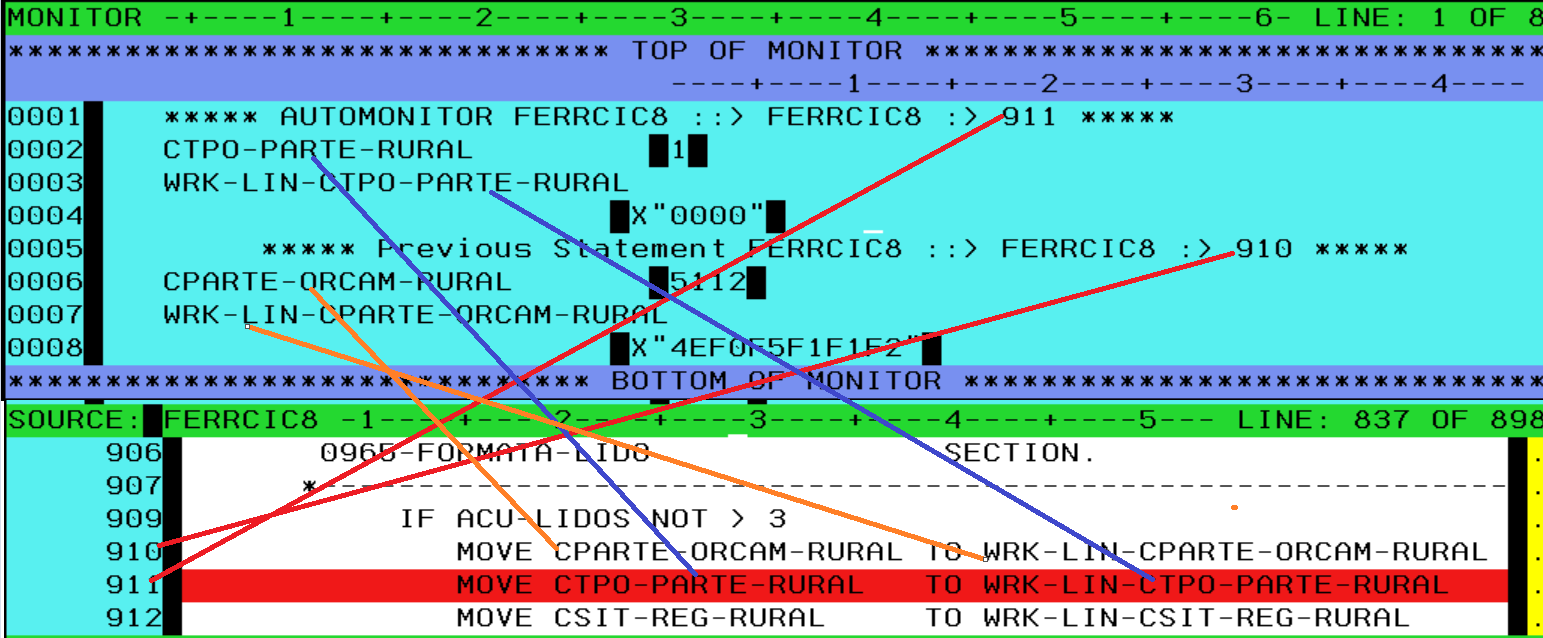


Entre com o comando SET AUTO ON BOTH<Enter>.

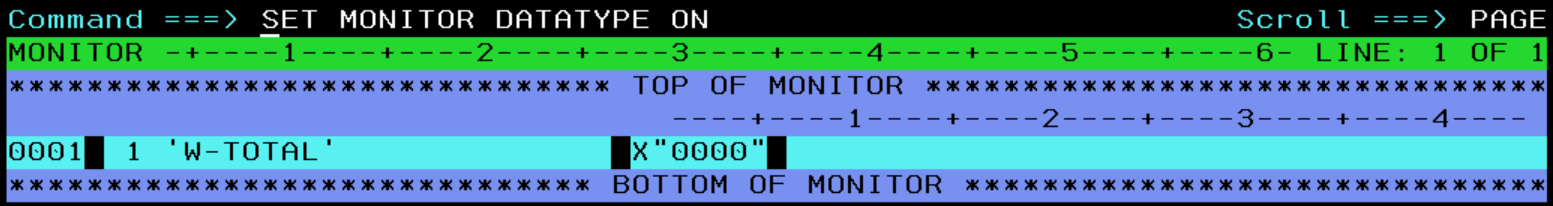
De acordo com as variáveis envolvidas no comando atual e anterior a janela de monitor vai mudando, mostra a variável do comando anterior e o atual antes da execução do comando.

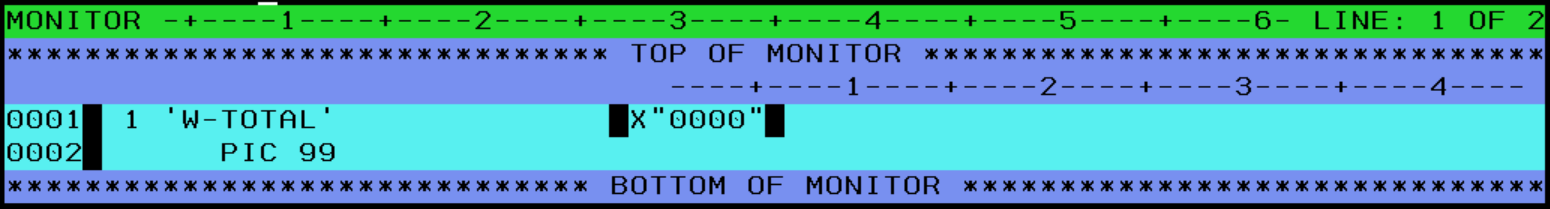


Mostra na parte superior da janela Monitor as variáveis da instrução corrente (linha 911) e na parte inferior as variáveis da instrução anterior (linha 910).



Para exibir o “tipo do dado” (PICTURE), utilize o comando SET MONITOR DATA TYPE ON.





Para retornar, utilize o comando SET MONITOR DATA TYPE OFF.

# **Lista de comandos e PF´s.**

|  |  |  |  |
| --- | --- | --- | --- |
| NOME | FUNÇÃO | TECLA | EXEMPLO |
| STEP | EXECUTAR PRÓXIMA INSTRUÇÃO IMEDIATAMENTE | PF02 |  |
| QUIT | ENCERRAR DEBUG | PF03 |  |
| AT / CLEAR | INSERIR / RETIRAR BREAK POINT | PF06 | AT 33  CLEAR AT 33 |
| FIND | PROCURAR STRING NO CÓDIGO | PF05 | FIND INICIO |
| ZOOM | EFETUA ZOOM NO CÓDIGO FONTE | PF10 | ZOOM |
| ZOOM LOG | EFETUA ZOOM NA JANELA DE LOG |  | ZOOM LOG |
| GO | EXECUTA O PROGRAMA ATÉ O PRÓXIMO BREAKPOINT | PF09 | GO |
| RUNTO | EXECUTA O PROGRAMA ATÉ A LINHA INFORMADA |  | RUNTO 33 |
| GO TO | DESVIA O FLUXO DA EXECUÇÃO PARA A LINHA APONTADA |  | GOTO 33 |
| LIST | MOSTRA O CONTEÚDO DE UMA VARIAVEL | PF04 | LIST AX-DISPLAY |
| MONITOR LIST | MOSTRA O CONTEÚDO DA VARIAVEL DURANTE TODA A EXECUÇÃO |  | MONITOR LIST AX-DISPLAY |
| SET MONITOR DATATYPE ON | MOSTRAR OS TIPOS DE DADOS DAS VARIÁVEIS (PIC DE DEFINIÇÃO) |  | SET MONITOR DATATYPE ON |

|  |  |  |
| --- | --- | --- |
| NOME | FUNÇÃO | EXEMPLO |
| POS | POSICIONA O FONTE NA LINHA nnn | POS 534 |
| QUALIFY RESET  ou  QUA RES | Posiciona o fonte na linha corrente. | QUA RES |
| STEP nn | Executa nn comando a partir da linha corrente, mostra a execução linha a linha com delay. | STEP 10 |
| JUMPTO nnn | Desvia a execução para o comanda da linha nnn, não executa a linha. | JUMPTO 346 |
| GOTO nnn | Desvia a execução para o comanda da linha nnn, e continua executando dali em diante. | GOTO 346 |
| AT CHANGE <var-name> | Coloca um Breakpoint quando a variável sofrer alteração. | AT CHANGE WRK-FILE-STATUS-ARQ1 |
| AT CHANGE <var-name> when <var-name> condição | Coloca um Breakpoint quando a variável sofrer alteração e a condição for verdadeira. | AT CHANGE WRK-FILE-STATUS-ARQ1 WHEN WRK-FILE-STATUS-ARQ1 =’10’ |
| LIST AT | Lista todos os Breakoints do programa | LIST AT |
| FINDBP | Encontra o proximo Breakpoint e posiciona a janela de fonte (Source) na linha. | FINDBP |
| CLEAR AT | Limpa todos os Breakpoints. | CEAR AT |
| WINDOW CLOSE <nome janela> | Fecha uma janela | WINDOW CLOSE LOG |
| WINDOW OPEN <nome janela> | Abre uma janela | WINDOW OPEN LOG |
| WINDOW SIZE nn <nome janela> | Altera o tamanho da janela especificada. | WINDOW SIZE 12 SOURCE |
| PANEL LAYOUT RESET | Restaura as janelas ao tamnho Default. | PANEL LAYOUT RESET |

**Assembler:**

|  |  |  |
| --- | --- | --- |
| NOME | FUNÇÃO | EXEMPLO |
| MEM <var-name> | Posiciona no endereço de memória da variável. | MEM IREC |
| MEM X’endereço’ | Posiciona no endereço de memória informado. | MEM ‘44E50’ |
| MEM %GRPrr-> | Posiciona no endereço de memória apontado pelo Registrador rr. | MEM %GRP15-> |
| LIST <var-name> | Lista o contéudo da variável na janela LOG. | LIST IREC |
| MON LIST <var-name> | M.ostra o conteúdo da variável na janela MONITOR. | MON LIST IREC |
| ZOOM MEM | Abre a janela MEMORY. | ZOOM MEM |
| LIST REG | Lista o conteúdo dos REGISTRADORES na janela LOG. | LIST REG |
| MON LIST REG | Lista o conteúdo dos REGISTRADORES na janela MONITOR. | MON LIST REG |

OBs.: para fechar a janela MEMORY aberta com o comando ZOOM MEM, utilize a <Pf10>.

# **Executando um programa Assembler com o Debug Tool no CICS.**

Para programas compilados em Assembler, é necessário ativara EXIT para programas “Non-LE”, para isso, limpe a tela do CICS (<Clear>) e digite a transação e parâmetros que segue: DTCXXO (DTCX: transação de ativação da EXIT, X:é o tipo da EXIT (no caso todas) e O: de On (no caso ativar).



Será exibida uma das duas mensagens abaixo que devem ser consideradas normal. 



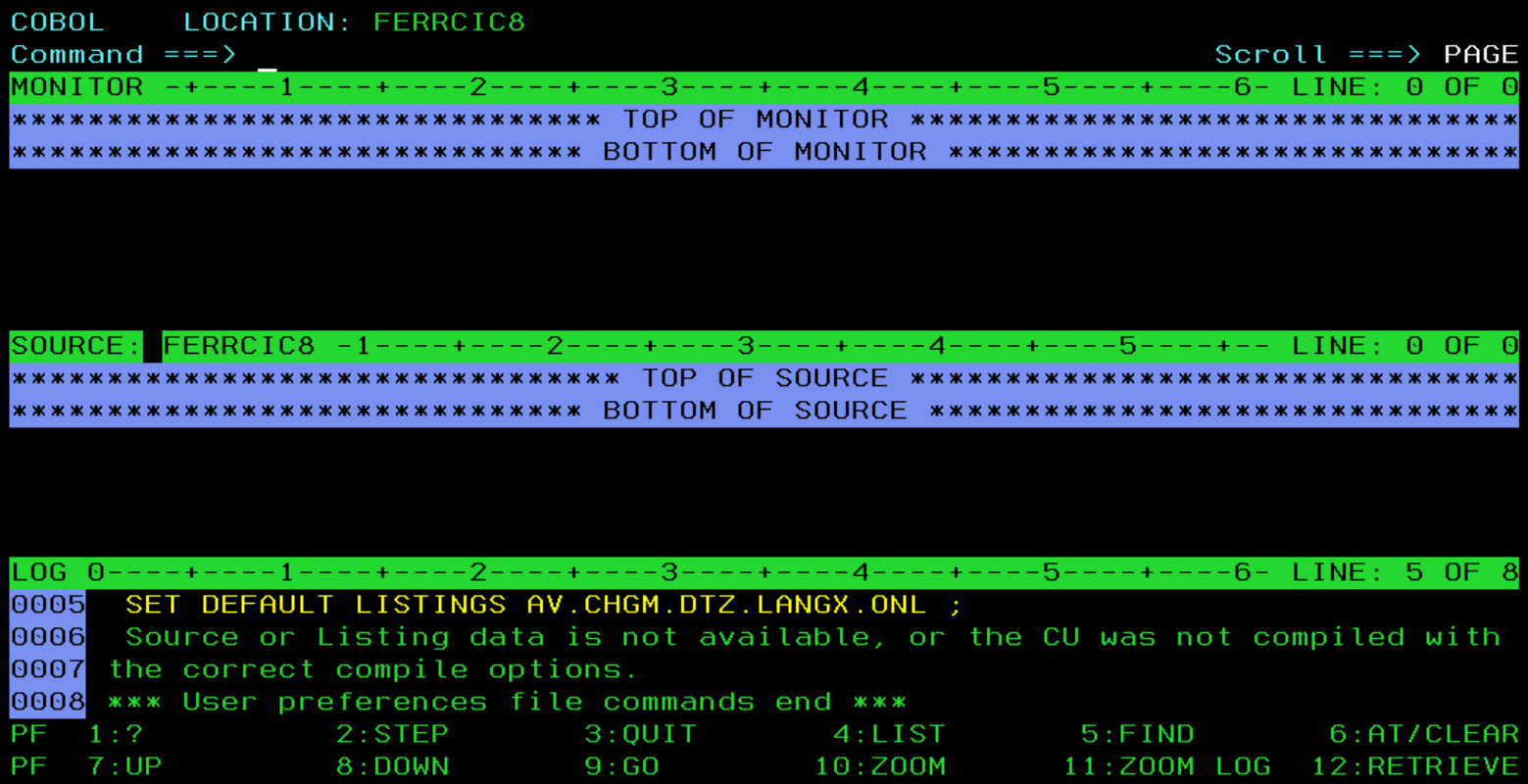
Significa que algum outro usuário já vez a ativação anteriormente.

Para desativar a EXIT, utilize: DTCXXF , onde o “F”: oFf (no caso desativar).



Após a ativação da EXIT, dê continuidade ao Debug seguindo os passos do item “[Executando um programa com Debug Tool no CICS](#_Executando_um_programa)”.

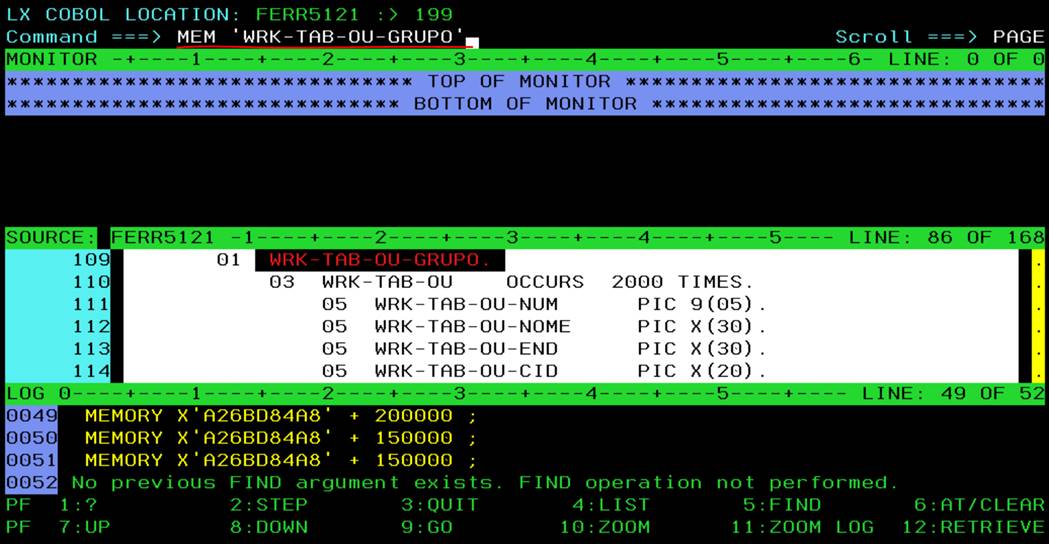
Será exibida a tela seguinte:



# **Apêndice I – Inspecionando variáveis na memória.**

Para verificar o conteúdo de variáveis “grandes” deve-se utilizar o comando MEMORY (MEM), conforme passos abaixo:

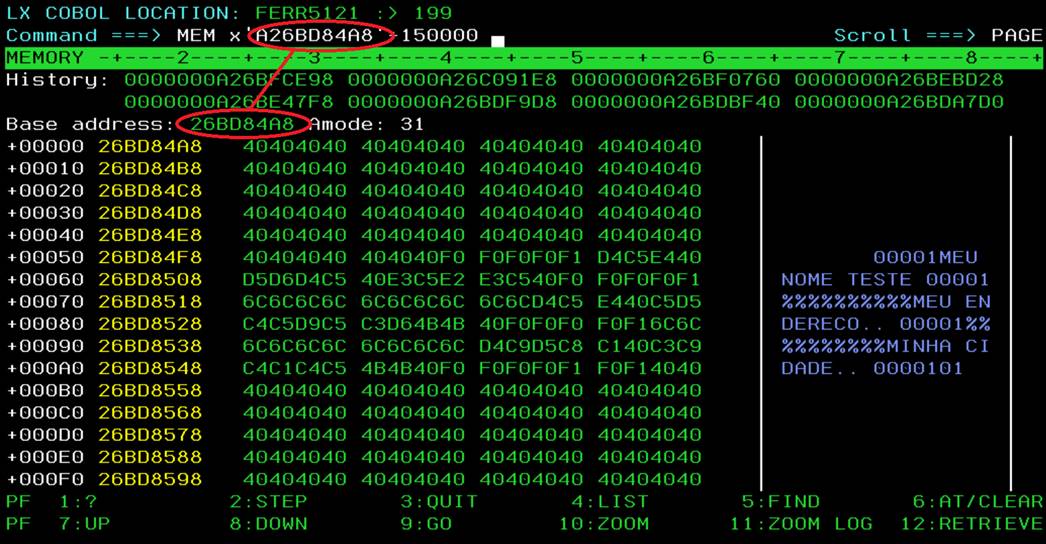
**Comando: MEM 'WRK-TAB-OU-GRUPO'**



**Comando: ZOOM MEM (para abrir a janela MEMORY)**



**Comando: MEM x'A26BD84A8'+150000 para deslocar dentro memória.**





**<PF10> (ZOOM) para fechar a janela MEMORY.**

# **Apêndice – Arquivos de configuração**

Arquivo: AV.CHGM.DTZ.LANGX.BAT e AV.CHGM.DTZ.LANGX.ONL

Alocação:

Organization . . . : PO

Record format . . . : VB

Record length . . . : 1562

Block size . . . . : 27998

Data set name type : LIBRARY

Conteúdo: Listagens dos programas para Debug, gerado pelo Changeman na compilação.

Arquivo: AD.DDS.A123456.DTSF

Alocação:

Organization . . . : PO

Record format . . . : VB

Record length . . . : 1280

Block size . . . . : 32760

Data set name type : LIBRARY

Conteúdo: Arquivo de Setup do Debug Tool, contém os JCL capturados para Debug, um por usuário.

Arquivo: AD.D493.READ.FERR.DTZ.PREF.BAT e AD.D493.READ.FERR.DTZ.PREF.ONL

Alocação:

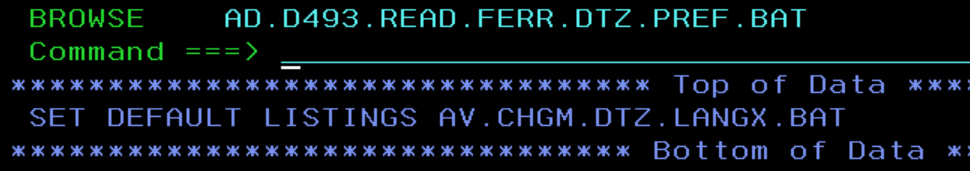
Organization . . . : PS

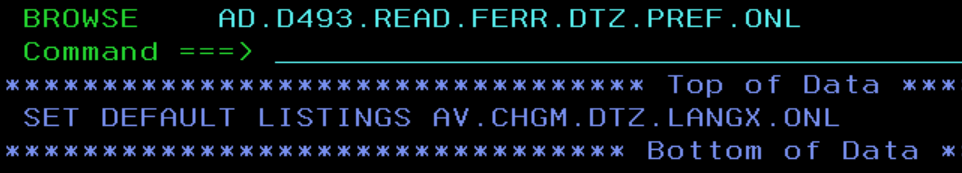
Record format . . . : FB

Record length . . . : 80

Block size . . . . : 27920

Conteúdo: Arquivo de Preferências do Debug Tool, pode conter comandos e deve apontar para a listagem do programa.





# **Apêndice IV – Link para manuais da IBM.**

**Quick Reference:** <ftp://public.dhe.ibm.com/software/iea/content/com.ibm.iea.debugt/debugt/12z/AdditionalResources/DebugToolV12ReferenceCard-01.pdf>

**IBM z/OS Debugger V14.1.9 User's Guide:**

<https://www.ibm.com/support/knowledgecenter/en/SSQ2R2_14.1.0/com.ibm.debugtool.doc/eqacus00.pdf>