**Sumário**

[**1.** **Preparando programa para Debug no Changeman.** 2](#_Toc47451023)

[**2.** **Preparando a transação para Debug no IMS.** 5](#_Toc47451024)

[**3.** **Preparando arquivo de configuração para a execução.** 9](#_Toc47451025)

[**4.** **Fazendo Logon no TIM-Terminal Interface Manager.** 13](#_Toc47451026)

[**5.** **Executando um programa com o Debug Tool no IMS.** 14](#_Toc47451027)

[**6.** **Painel padrão do Debug Tool.** 17](#_Toc47451028)

[**7.** **Carregando a listagem de um programa.** 19](#_Toc47451029)

[**8.** **Descrição das PF´s.** 22](#_Toc47451030)

[**9.** **Lista de comandos e PF´s.** 31](#_Toc47451031)

[**10.** **Encerrando o Debug de uma transação IMS.** 34](#_Toc47451032)

[**11.** **Apêndice I – Inspecionando variáveis na memória.** 35](#_Toc47451033)

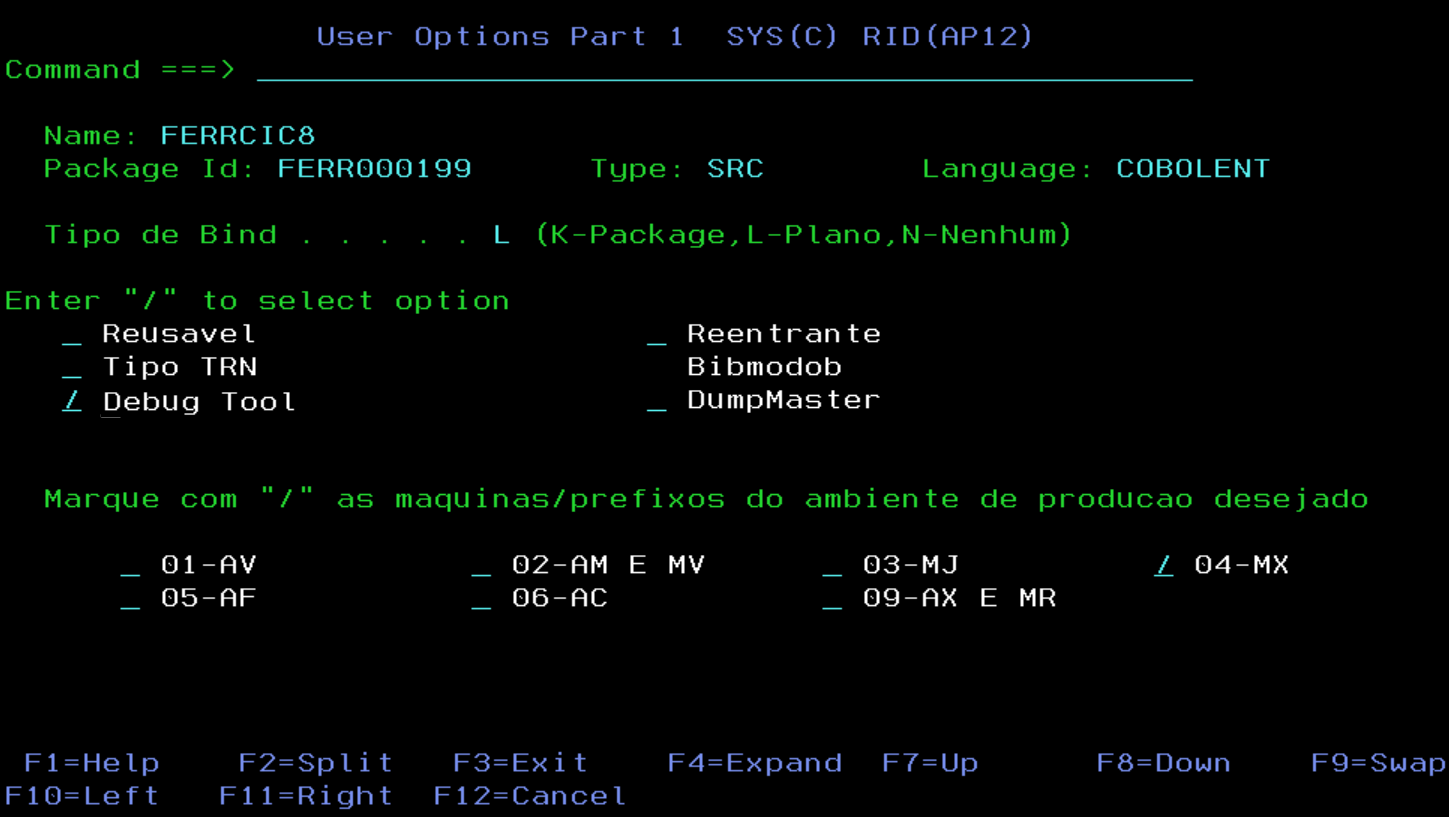
[**12.** **Apêndice – Arquivos de configuração** 37](#_Toc47451034)

[**13.** **Apêndice IV – Link para manuais da IBM.** 38](#_Toc47451035)

**IBM – Debug Tool - IMS**

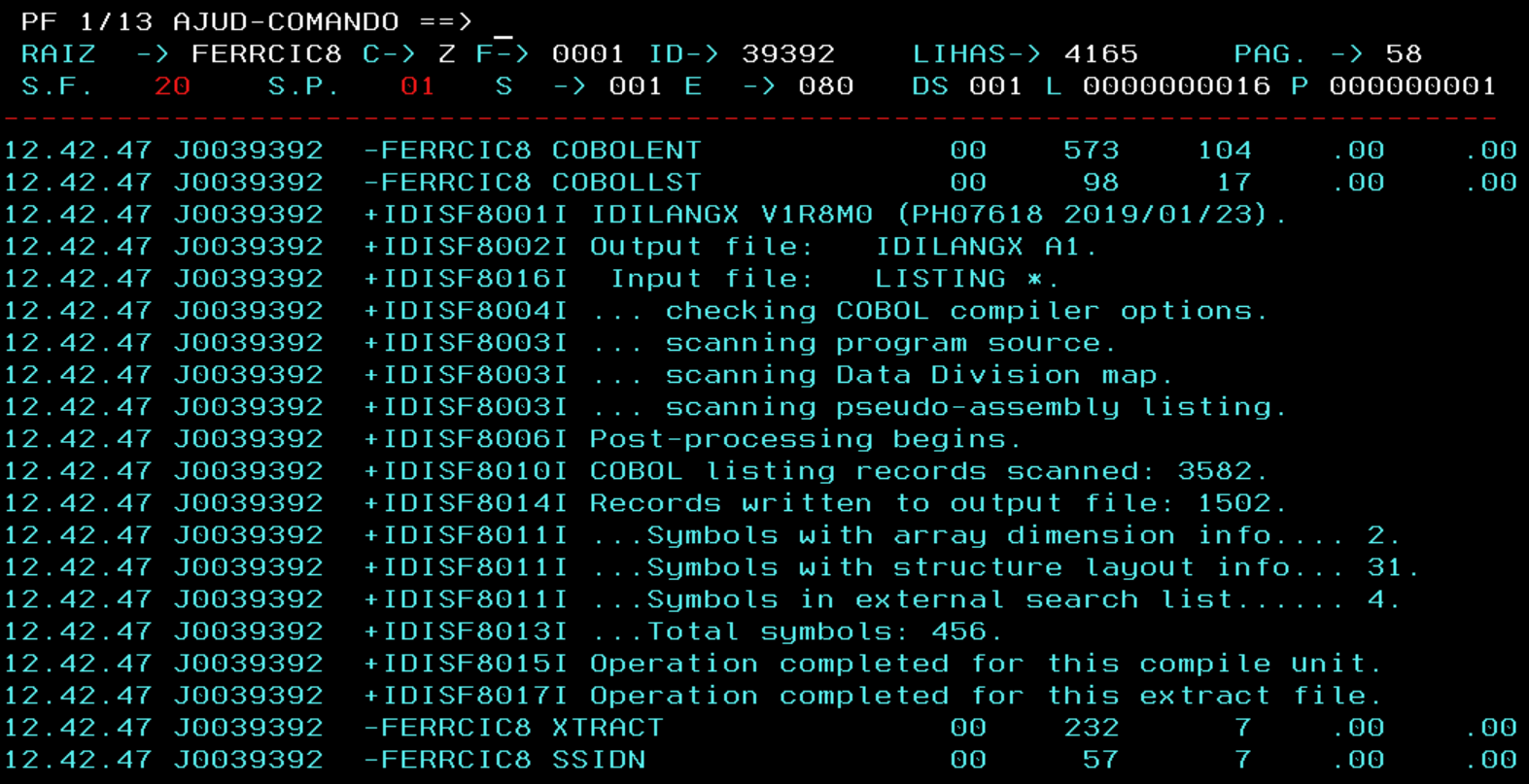
# **Preparando programa para Debug no Changeman.**

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.

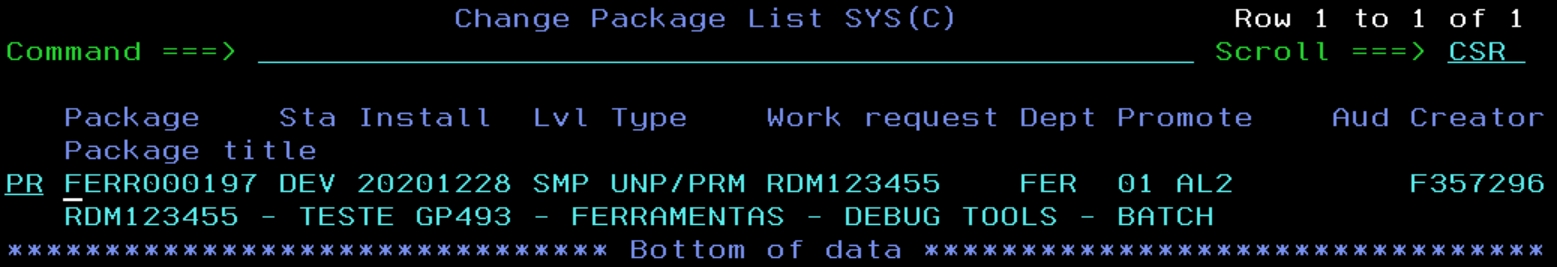


Na execução do JOB de compilação a listagem do programa será direcionada para o arquivo AV.CHGM.DTZ.LANGX.BAT que será utilizada no processo de Debug.

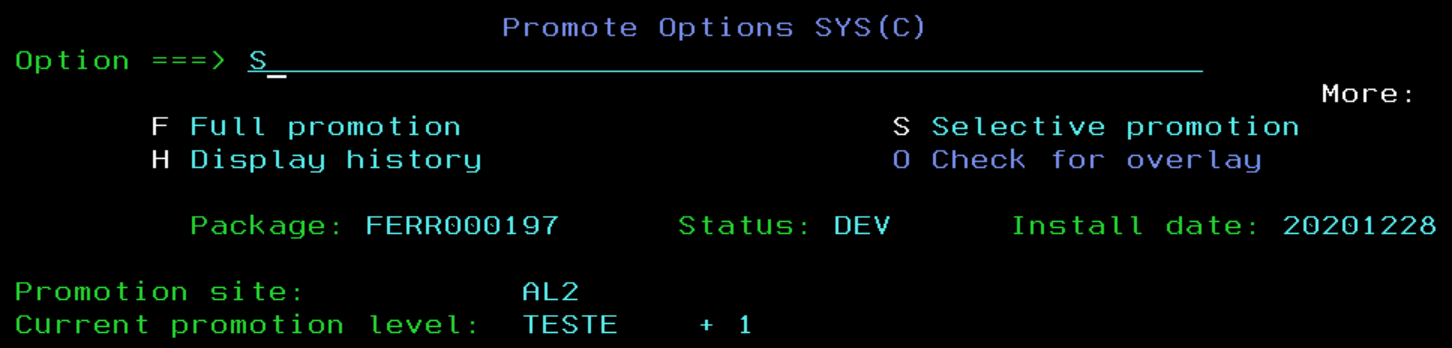
Na SYSOUT do JOB de compilação pode ser encontrada algumas mensagens IDISF80xxI que são geradas pelo Extrator do Debug.

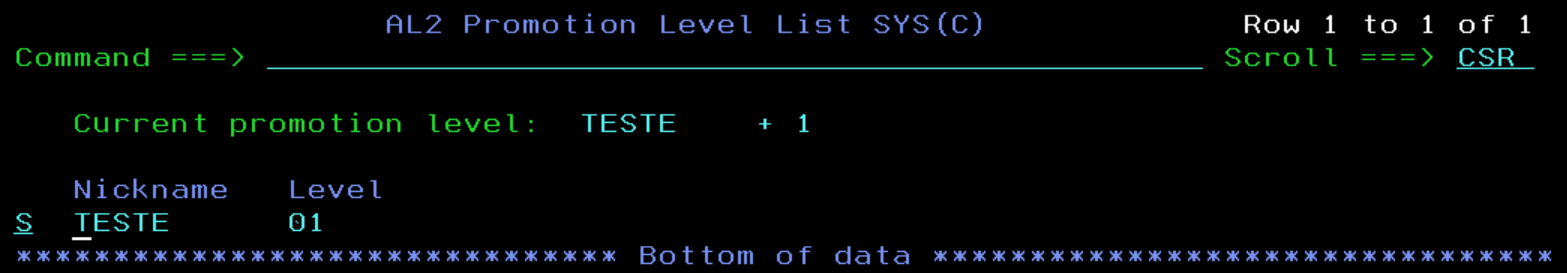


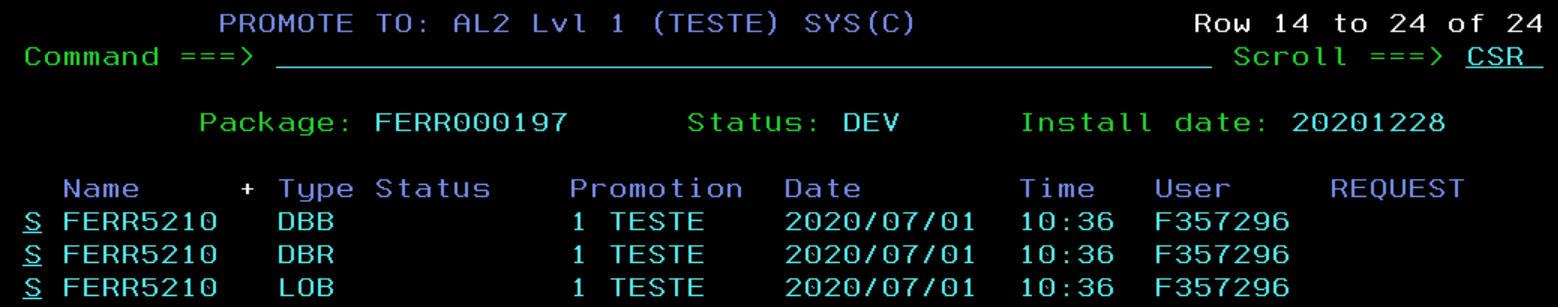
Após a compilação, devemos realizar o processo de **PROMOTE** para o ambiente de TU.











# **Preparando a transação para Debug no IMS.**

Faça o logon no IMS onde a aplicação irá executar.



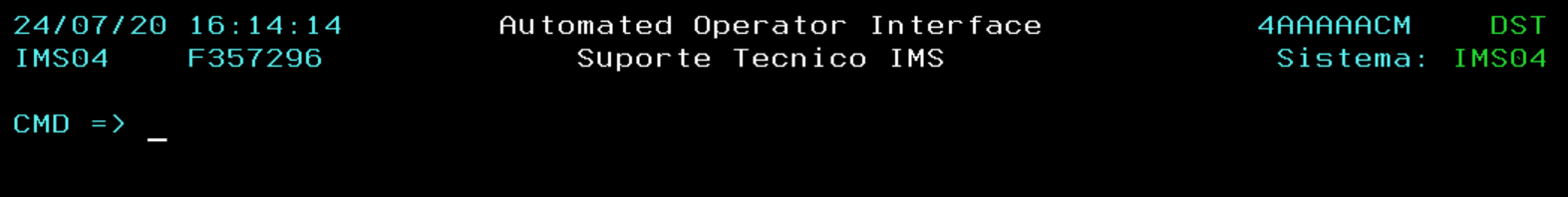


Tecle <Pf6> e faça o Sign-On com a sua chave e password.

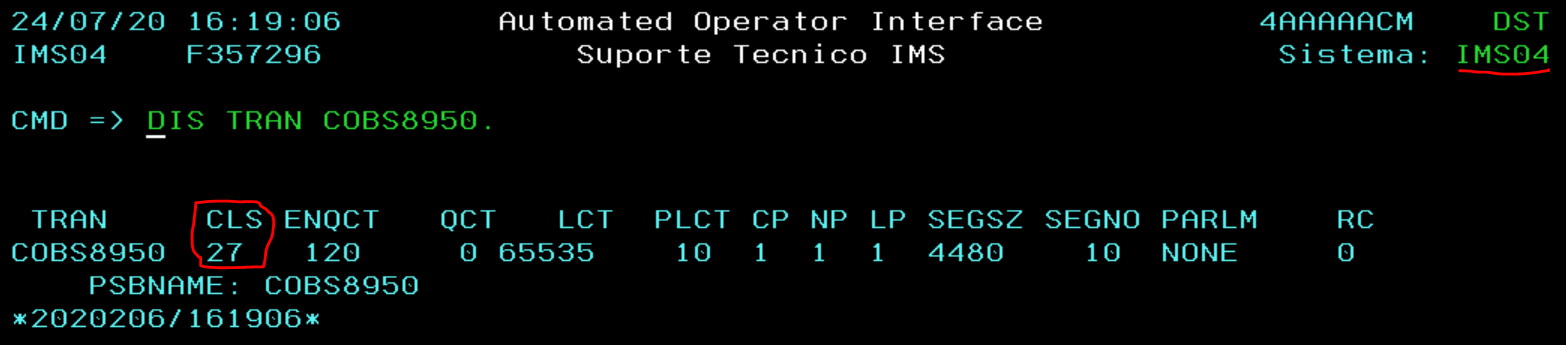


Entre na transação STIDST.

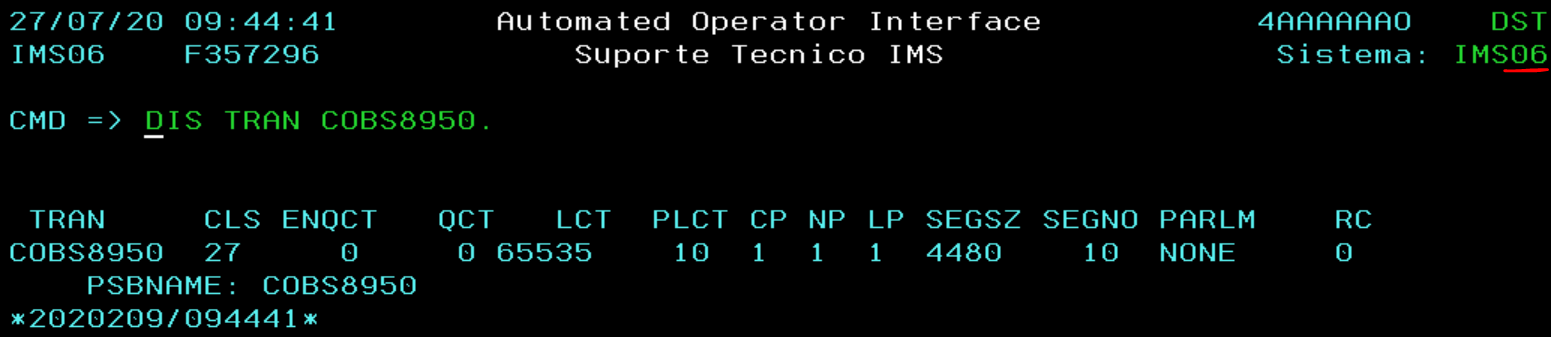




Vamos verificar em qual classe do IMS a transação está assinalada para rodar, através do comando: “DIS TRAN tttttttt”, onde: tttttttt=código da transação.

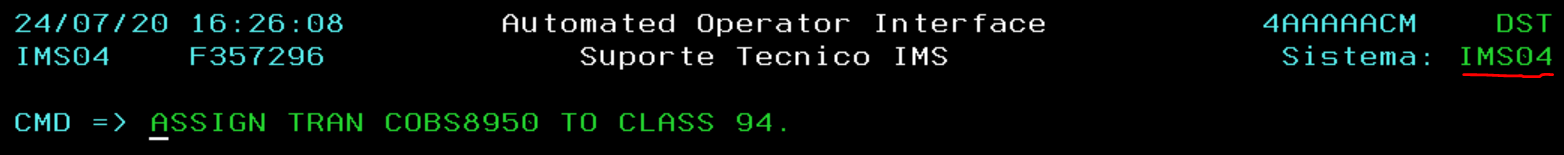


Temos que verificar nos dois IMSs, no IMS04 e no IMS06 (por ser IMSPlex).

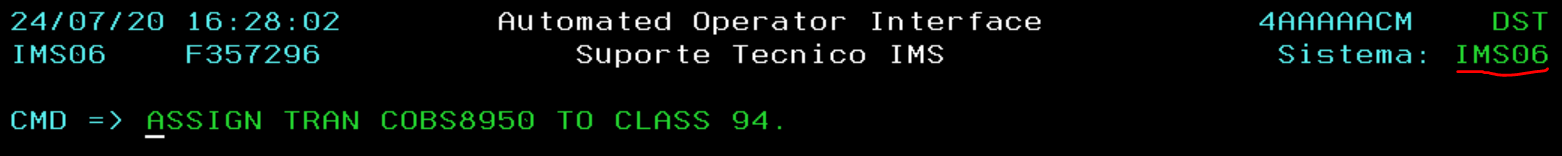


Neste caso, a transação está na sua classe “original”, ou seja “27”.

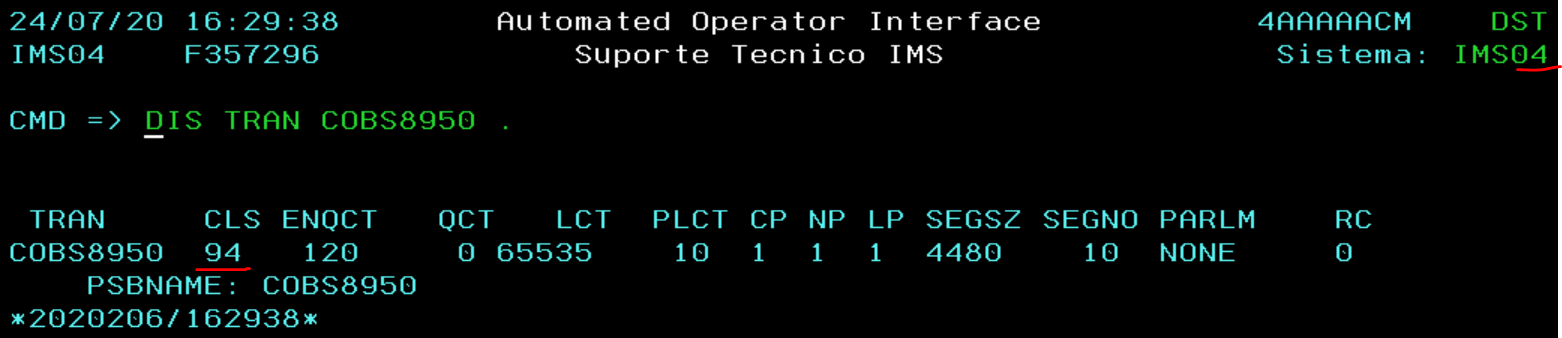
O próximo passo será alterar a classe da transação para “94”, que é a classe que foi designada para Debug de transações IMS, para isto iremos utilizar o comando: “ASSIGN TRAN tttttttt TO CLASS 94”, ONDE: tttttttt é o código da transação.



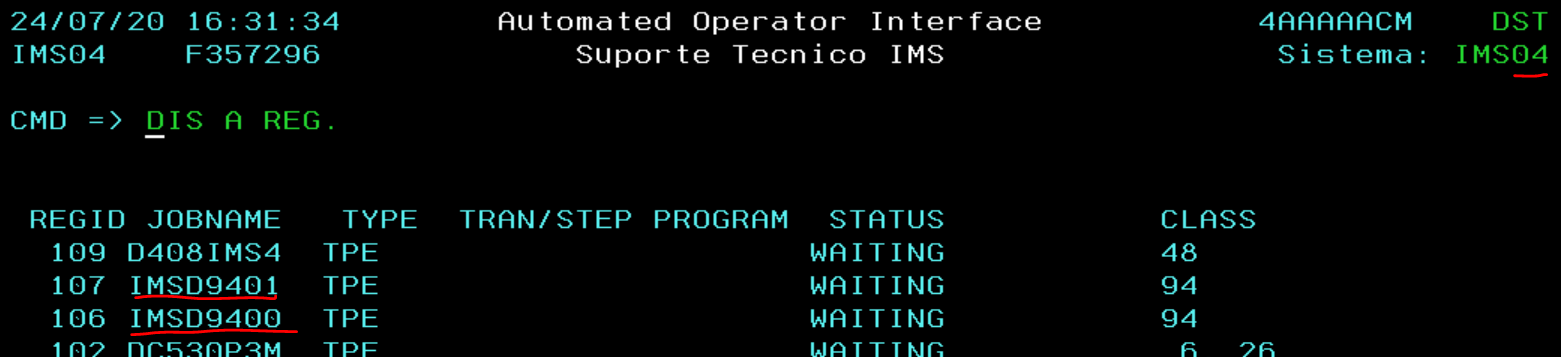
Temos que alterar a classe no IMS06 também.



Para verificar, utilize novamente o comando: “DIS TRAN tttttttt”, onde: tttttttt=código da transação.



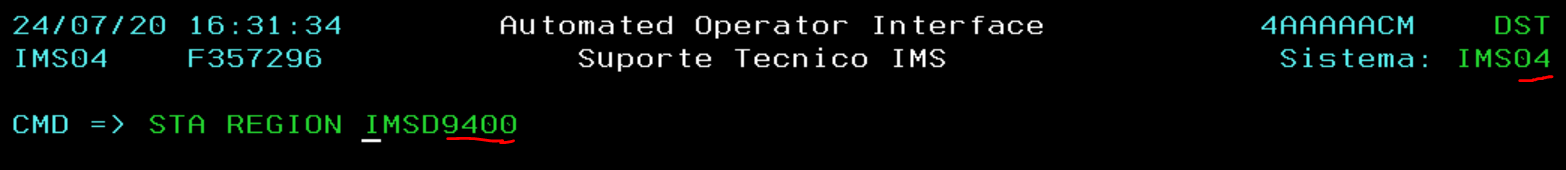
Em seguida, vamos verificar se as Regions IMS que atendem a classe “94” estão ativas, utilizando o comando “DIS A REG”, neste caso somente no IMS04.



O Jobname das regions da classe “94” seguem o padrão IMSD94nn, onde: nn=sequência numérica.

Caso exista pelo menos uma Region atendendo a classe “94”, pule o próximo passo, onde descrevemos como fazer o Start de uma Region.

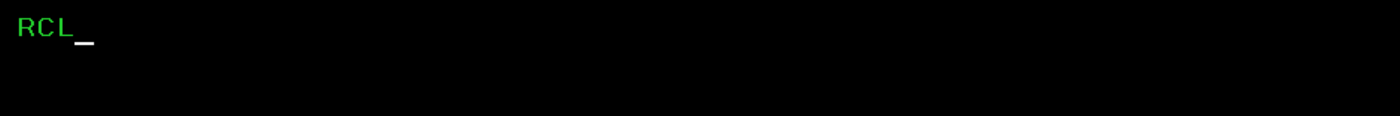
Para fazer o START de uma Region da classe “94”, utilizamos o comando: “STA REGION IMSD94nn”, onde: nn=sequência numérica.



Aguarde alguns segundos (pelo menos 30 segundos) e verifique se a Region foi “Startada” com o comando: “DIS A REG”.

Os procedimentos de preparação da transação foram concluídos, podemos sair do IMS.

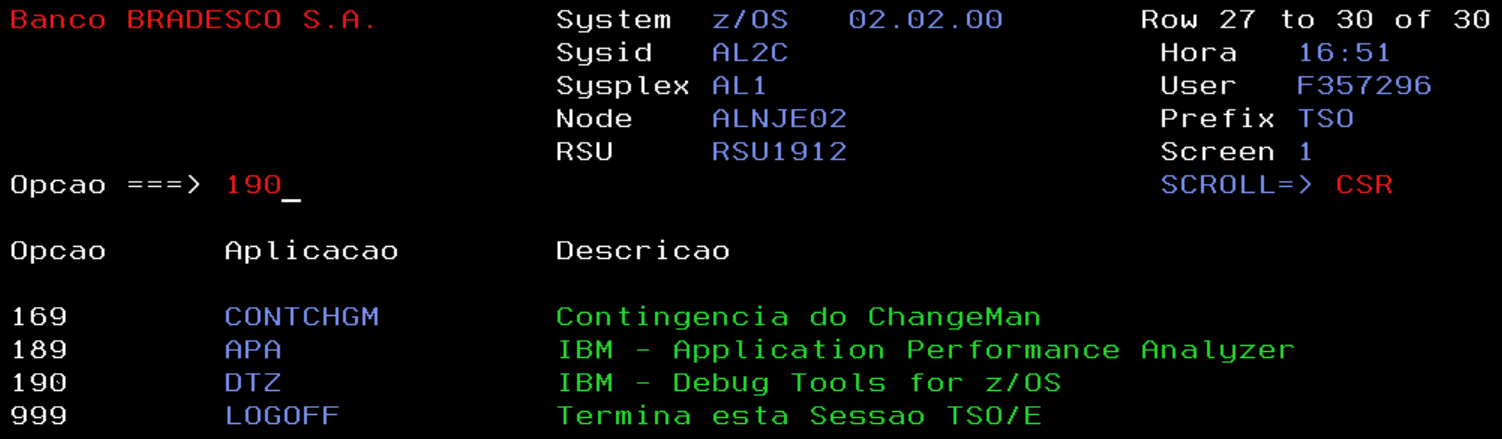
<Clear>



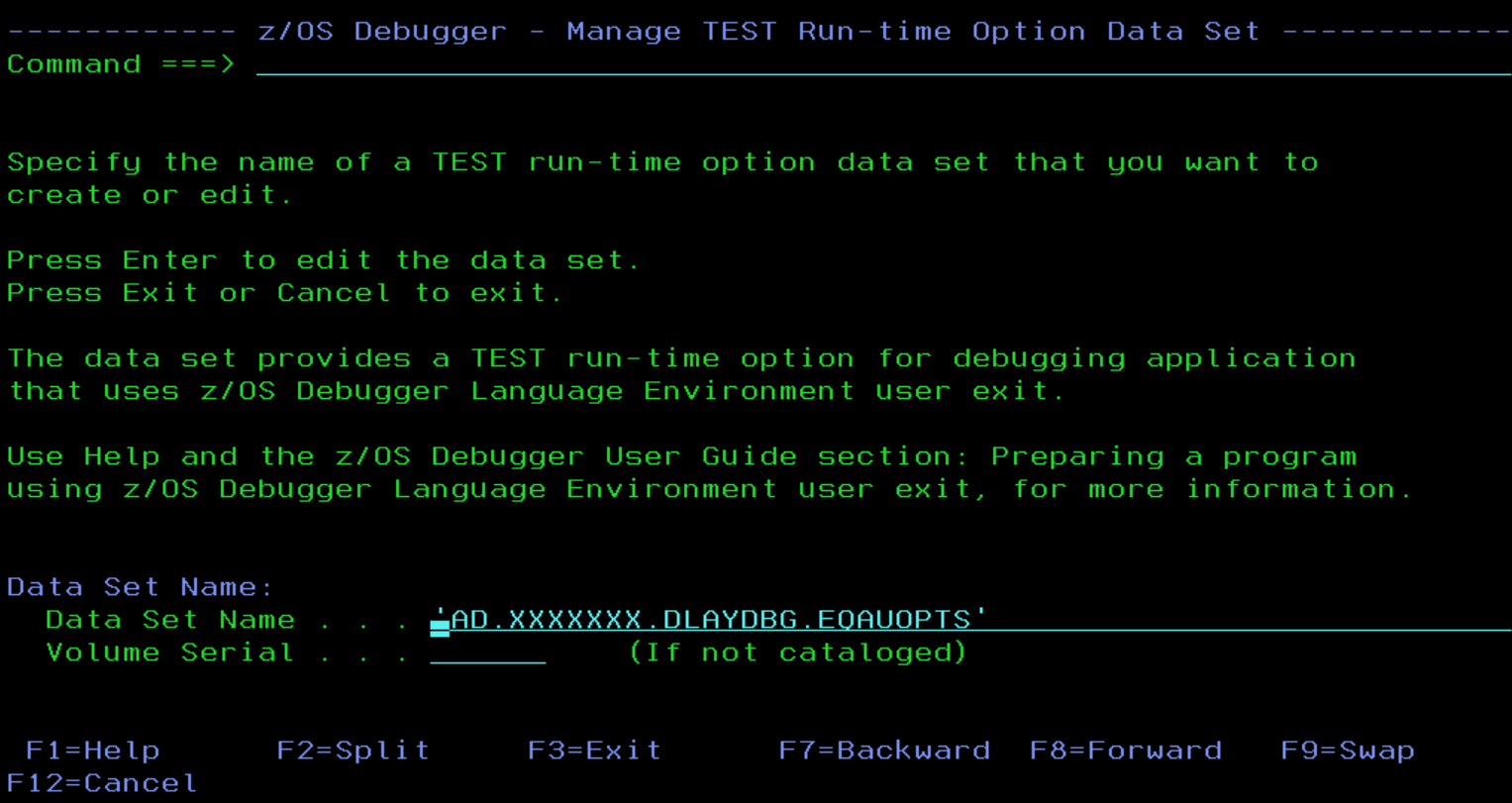
# **Preparando arquivo de configuração para a execução.**

Vamos entrar no TS48, na Opção 190 (DTZ - IBM - Debug Tools for z/OS).





Iremos utilizar a opção 6 ( z/OS Debugger User Exit Data Set).



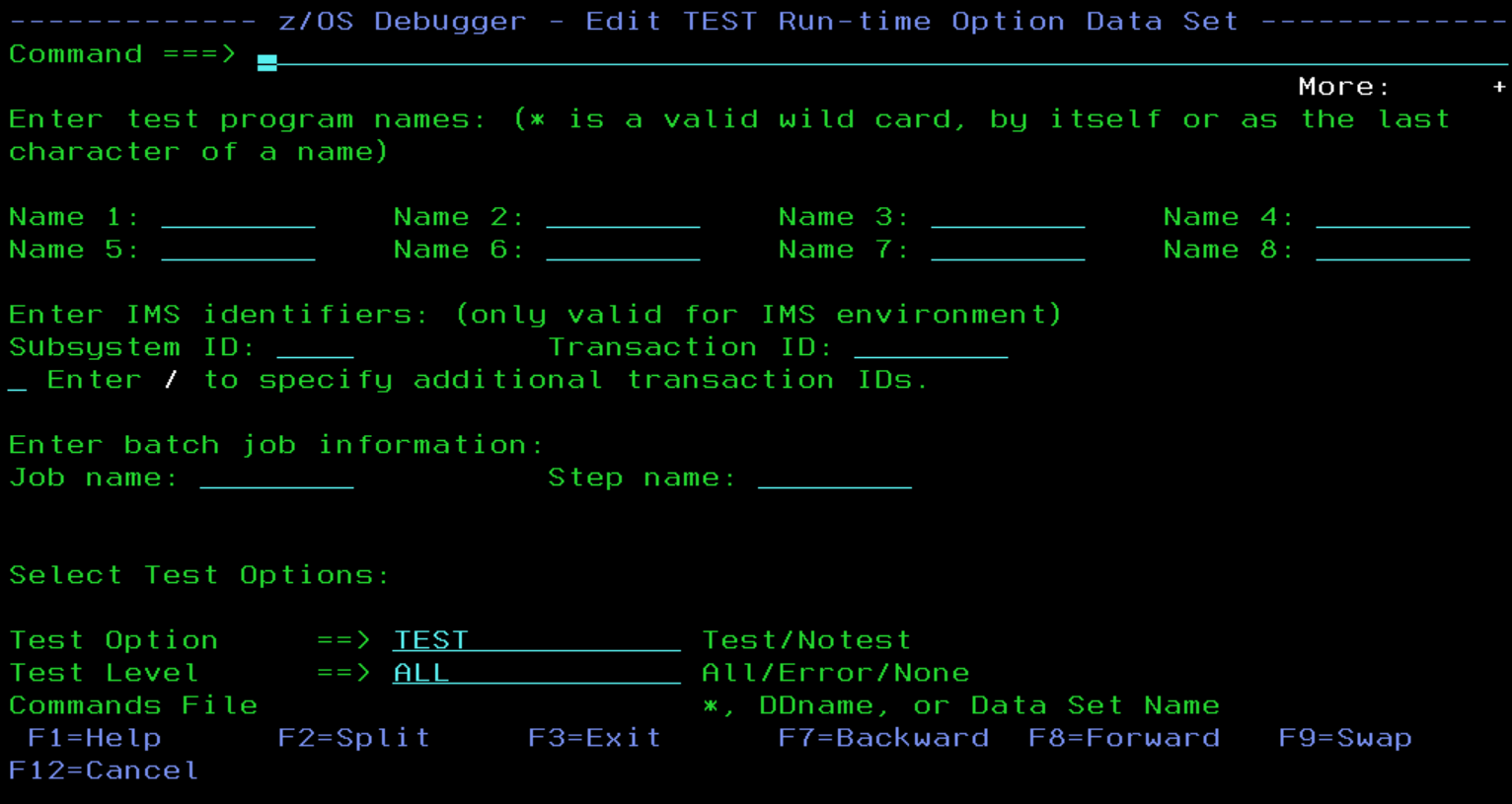
Nesta tela, deverá ser informado o arquivo no campo “Data Set Name” no padrão: 'AD.xxxxxxx.DLAYDBG.EQAUOPTS', onde: xxxxxxxxxx = chave do usuário que fará o Debug.

Se o arquivo não existir, será exibida a tela abaixo, para informar os parâmetros de alocação, simplesmente dê <Enter>, assumindo os valores “Default”.

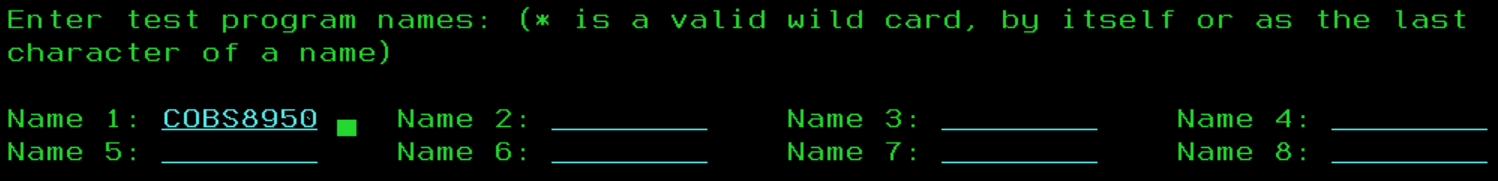


<Enter> para continuar.

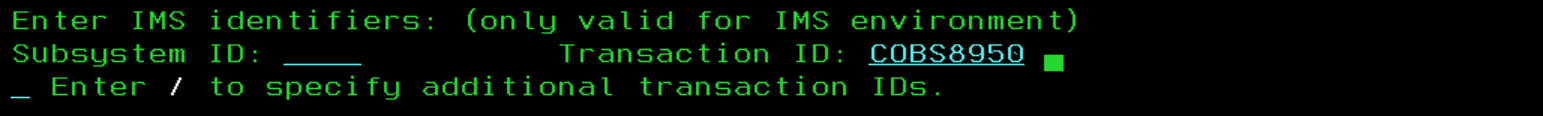
Será exibida a tela abaixo, onde deveremos informar os parâmetros de Debug.



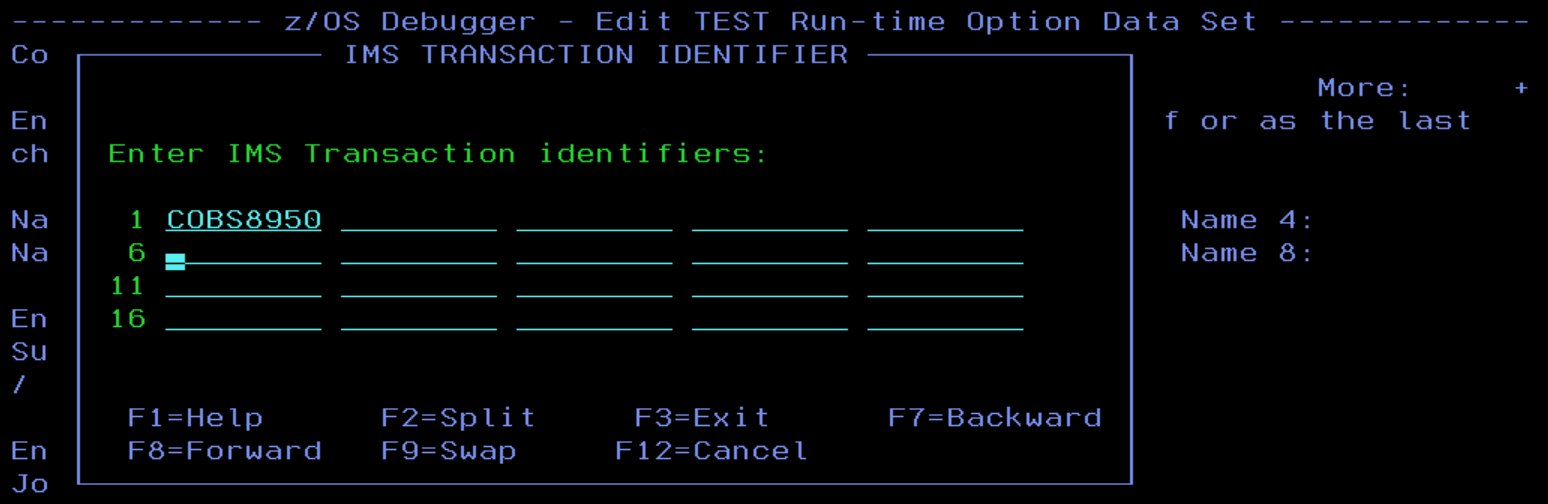
Informe o(s) programa(s) que serão “Debugados”.



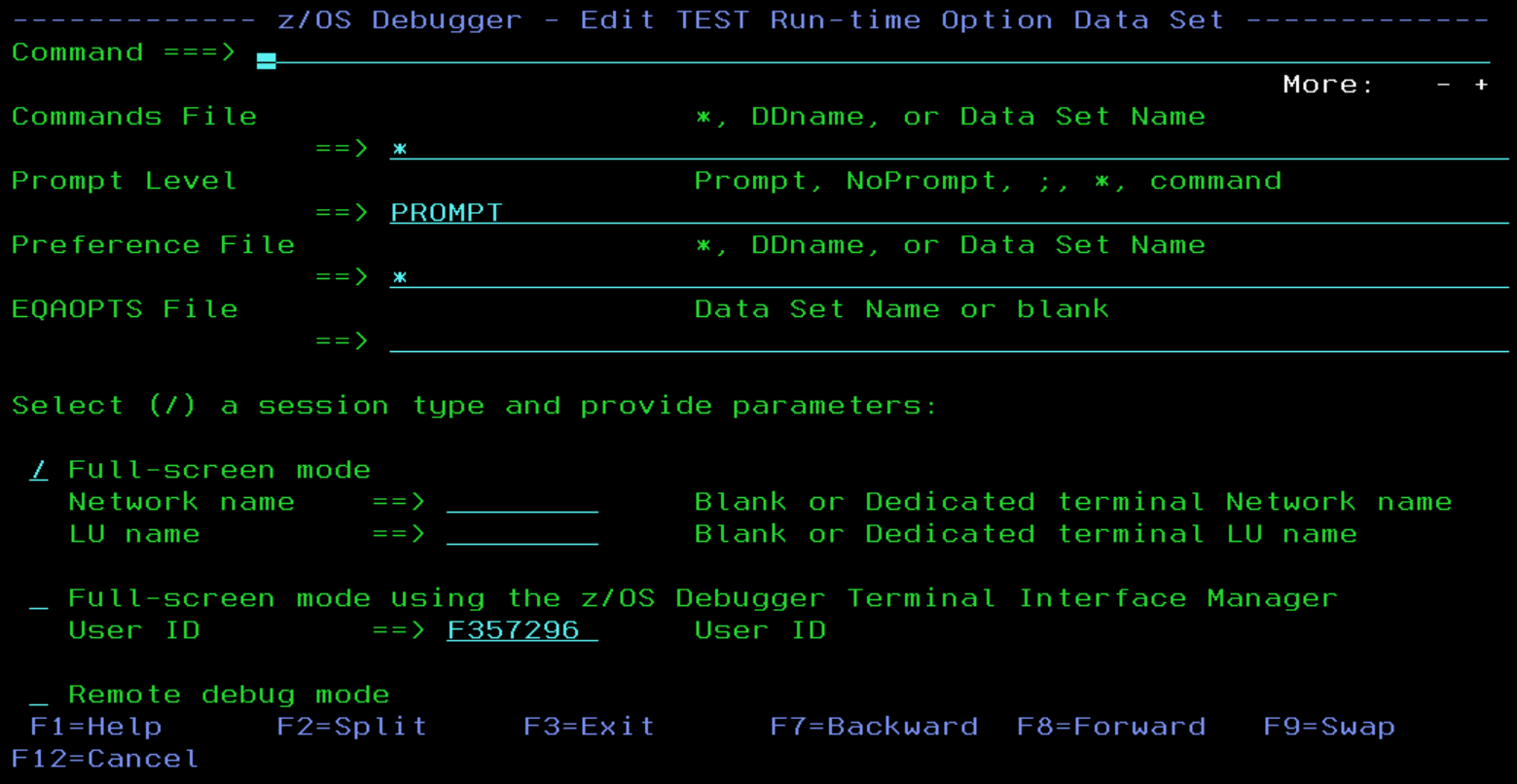
Informe a transação:



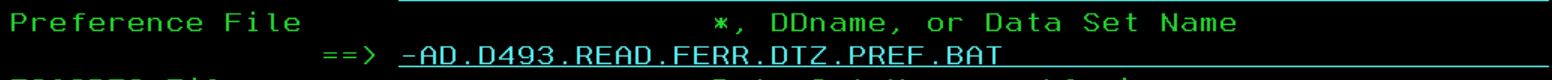
Caso seja necessário informar mais de uma transação, selecione com “/” o campo “ Enter / to specify additional transaction IDs.” Que será aberto uma janela.



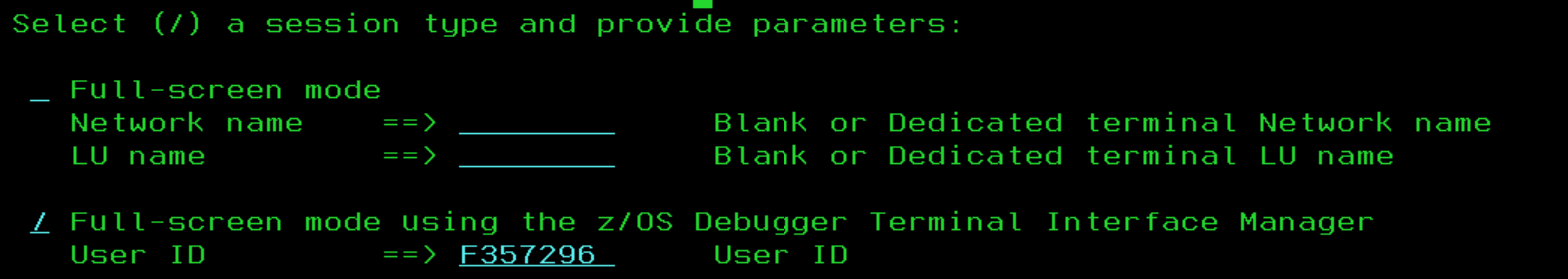
Utilize <Pf3> para sair da Janela e em seguida <Pf8> para paginar e informar os demais parâmetros.



Informe o arquivo **-**AD.D493.READ.FERR.DTZ.PREF.BAT, atenção sempre com o sinal “**-**“ na frente e **sem aspas**, no campo “Preference File”.



Selecione com “/” o campo “Full-screen mode using the z/OS Debugger Terminal Interface Manager” e informe a chave do usuário no campo “User ID”.



**Não esqueça de retira a “/” da opção “Default” no campo “Full-screen mode”.**

Digite <Enter> para confirmar e <Pf3> (3 vezes) para sair do Debug Tool.

Os procedimentos de preparação do arquivo de configuração foram concluídos, podemos sair do TSO.

# **Fazendo Logon no TIM-Terminal Interface Manager.**

Flavio:

1. Entrar no ACCTER.
2. Estando no ACCTER, informar na linha de comando: EQASESSM.

Antes de executar o programa é necessário entrar em outra sessão de terminal e fazer o login no TIM, seguindo os passos:

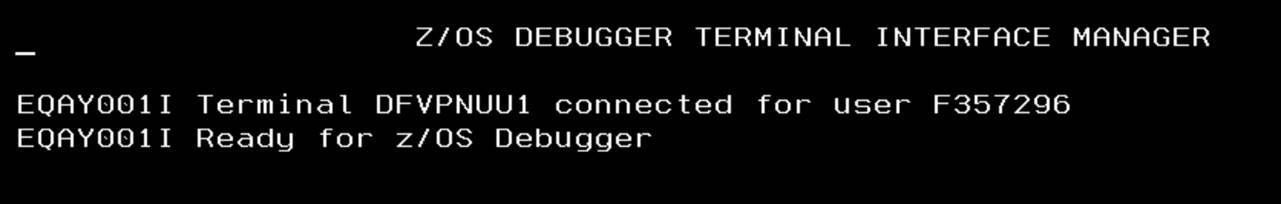
Logon applid(EQASESSM) ou Logon applid(braddsv.EQASESSM) ou ACCTER EQASESSM



Entrar com sua chave e Password.



Deixar nesta tela e retornar a outra sessão de terminal:



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

Entre no IMS04.

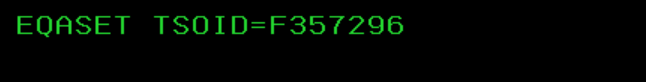




Tecle <Clear> nesta tela e entre com o comando “/TEST MFS”

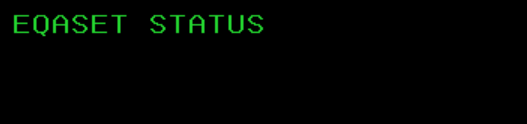


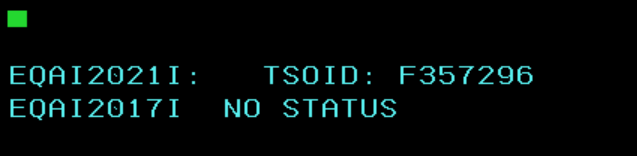
Antes de executar o programa, é necessário assinala um usuário para a sessão de debug, vamos utilizar a transação EQASET para isto, digite: “EQASET TSOID=cccccccc”, onde: cccccccc = chave do usuário.



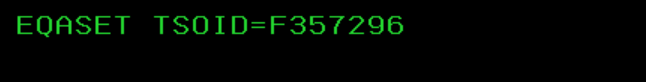


Para verificar se o usuário foi assinalado, utilize o comando “EAQSET STATUS”.

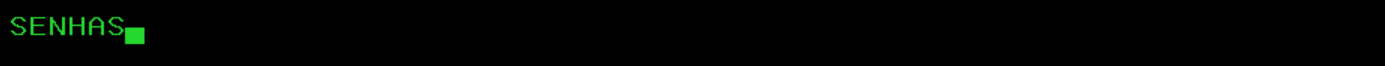




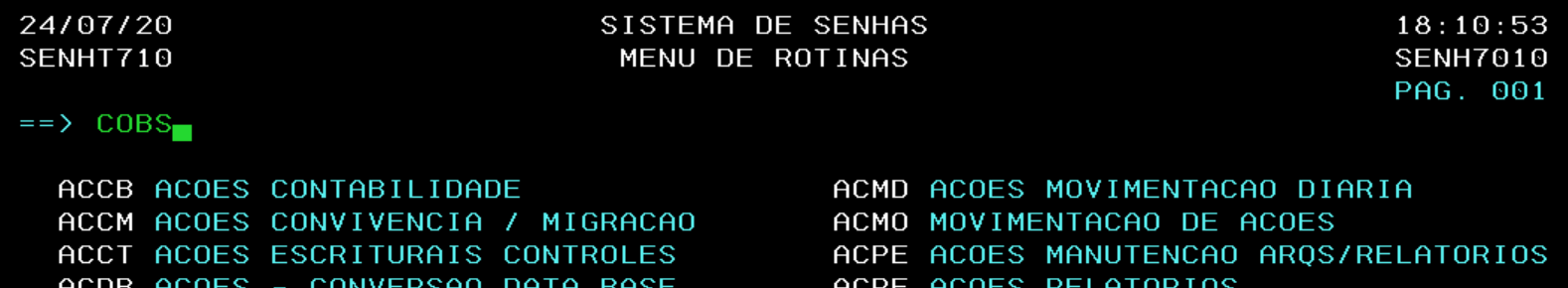
Note que é muito importante a “chave do usuário” estar “alinhada” em três “locais”.

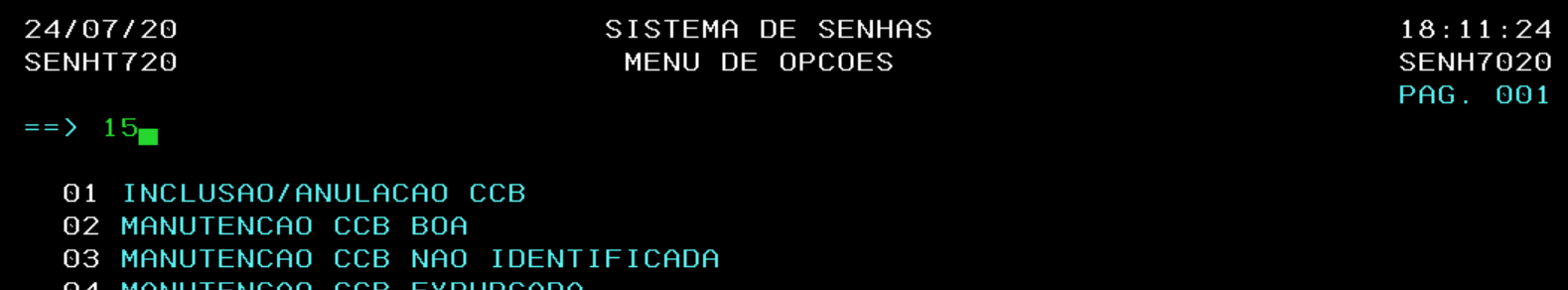
1. No “EQASET”: 
2. No “TIM”: 
3. No TSO, Opção 190, 6: 

Agora é só executar a sua transação.

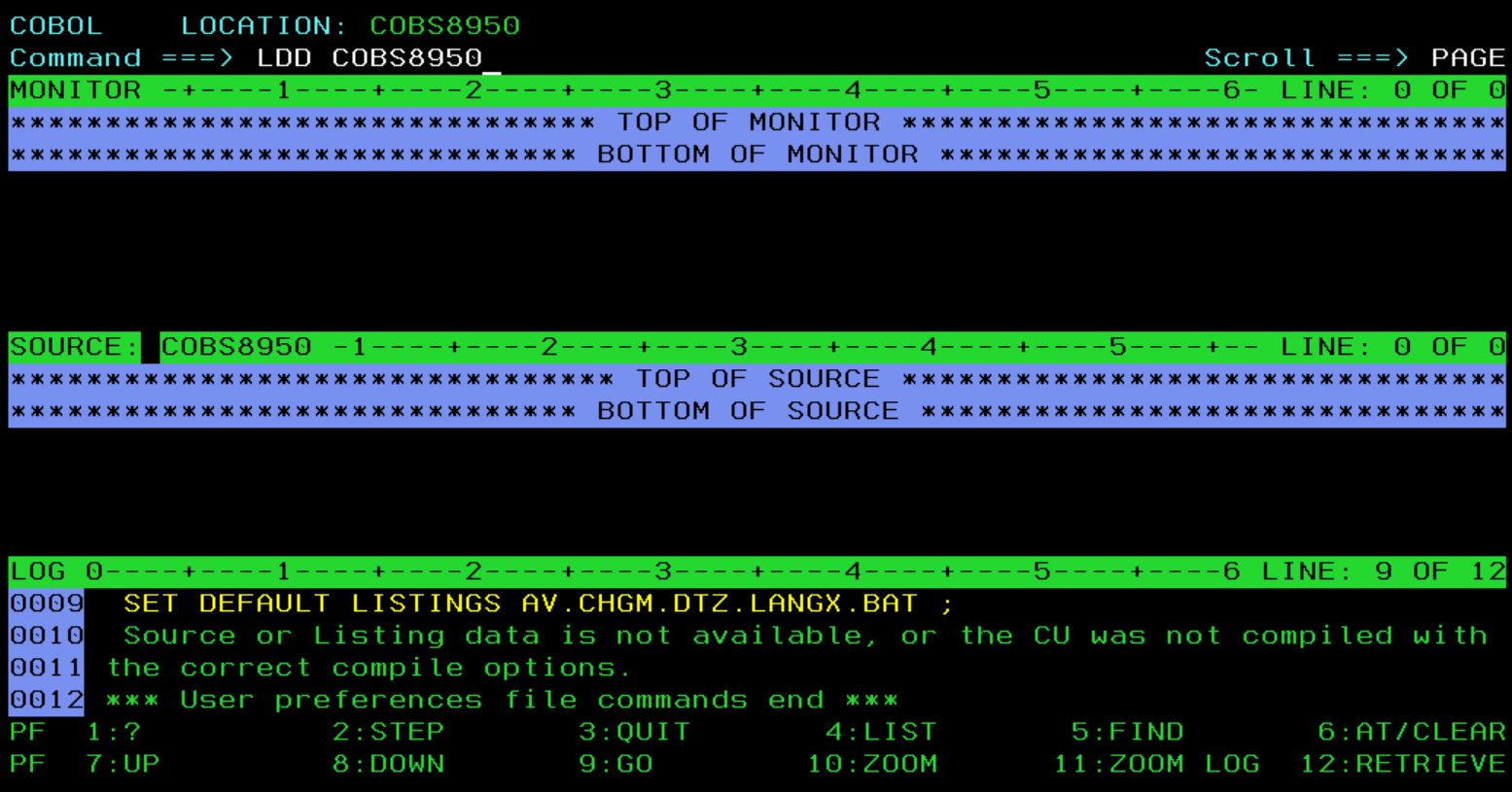




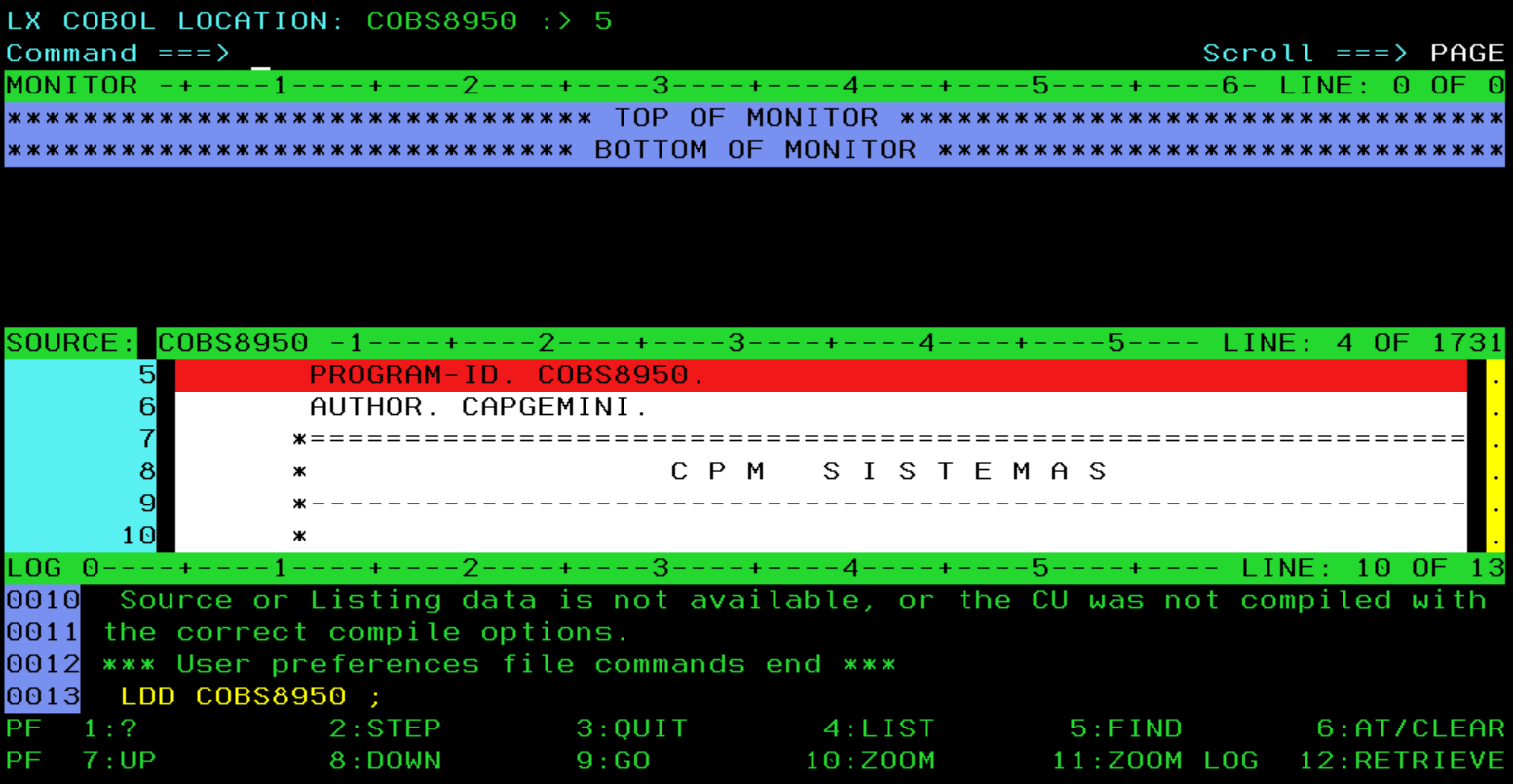




Após a escolha da opção o programa será interceptado e exibida a tela de Debug na sessão logada no TIM (z/Os Debugger Terminal Interface Manager).



Digite: LDD pppppppp, onde: pppppppp=código do programa.

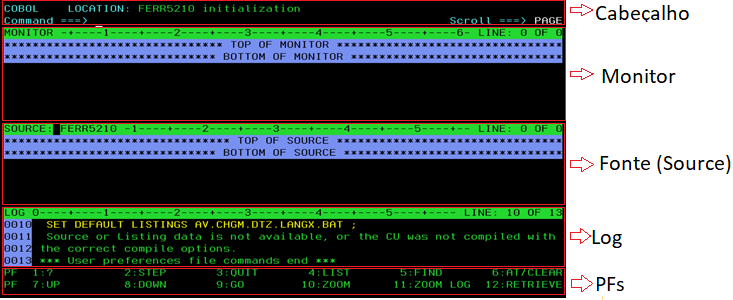


Dar continuidade no Debug utilizando os comandos e PFs descritos nos itens ,[Painel padrão do Debug Tool.](#_Painel_padrão_do) [Carregando a listagem de um programa.](#_Carregando_a_listagem) e [Descrição das PF´s](#_Descrição_das_PF´s.).

**Observação:** após encerrar as atividades de Debug da transação IMS é muito importante fazer a “limpeza” dos recursos alocados, evitando travamento das suas transações e liberando recursos para outros usuário, utilize as instruções descritas no item [Encerrando o Debug de uma transação IMS.](#_Encerrando_o_Debug)

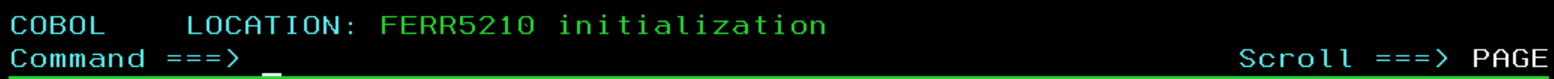
# **Painel padrão do Debug Tool.**

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.



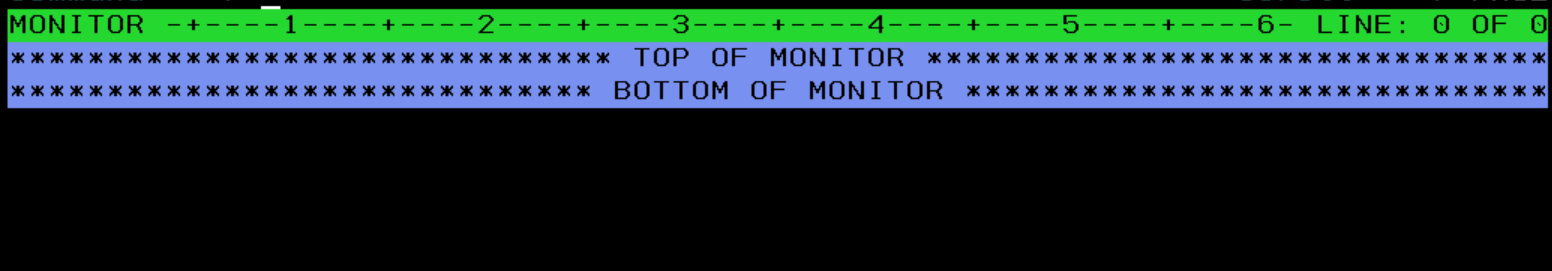
**Área de cabeçalho:**

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.



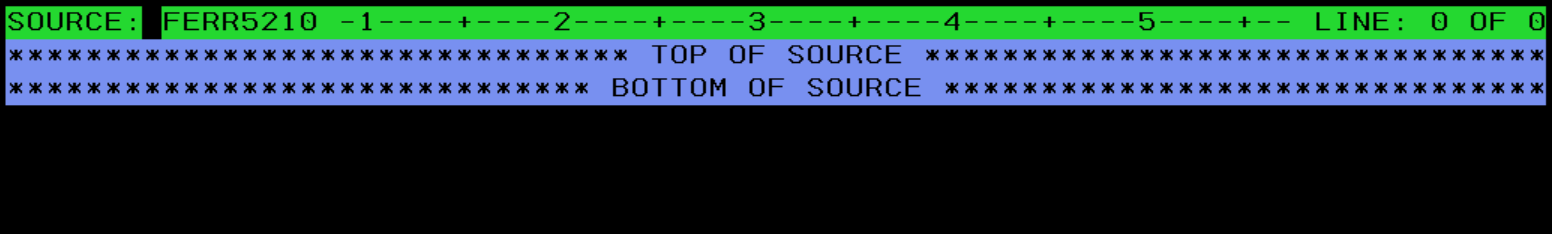
**Janela Monitor:**

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”.



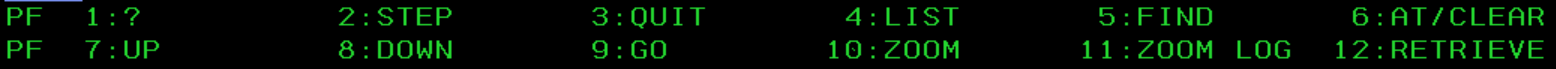
**Janela Log:**

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.

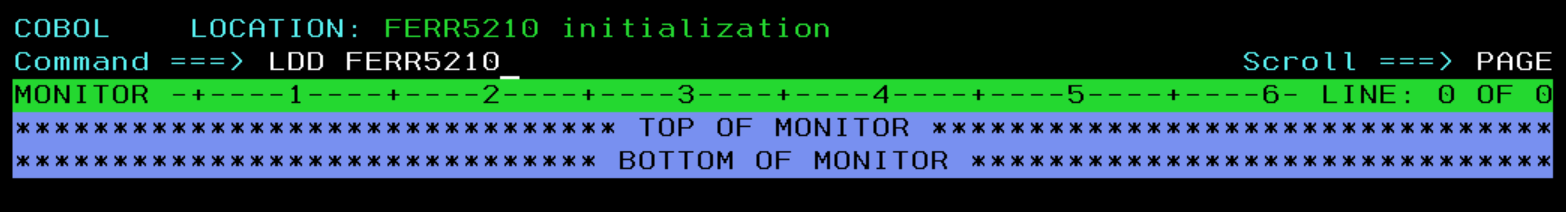


Mais adiante descreveremos as funções da PF´s e alguns outros comandos.

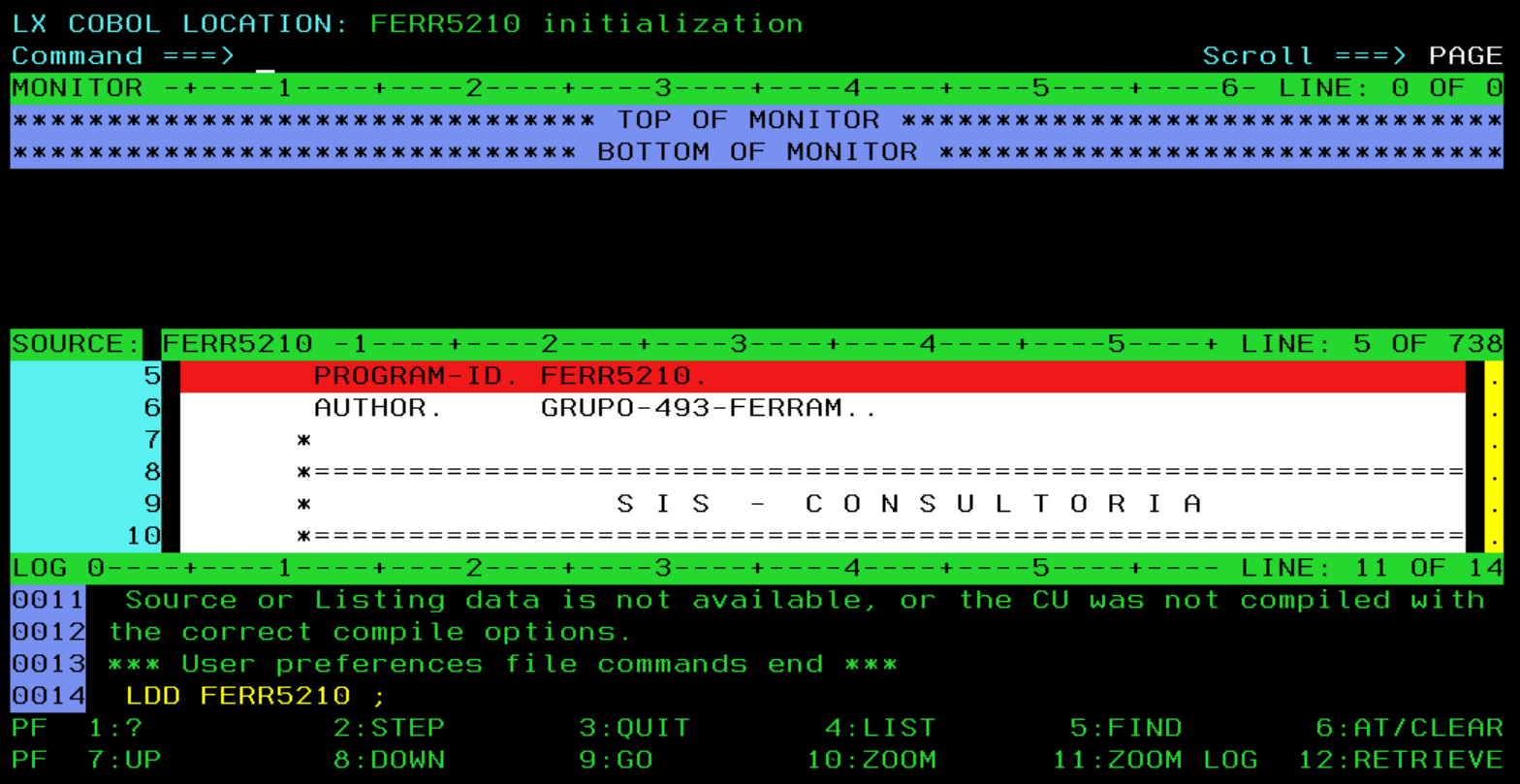
# **Carregando a listagem de um programa.**

Para as linguagens Cobol II, MVS Cobol e Enterprice Cobol v3.4 é necessário fazer a carga da listagem do programa 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.

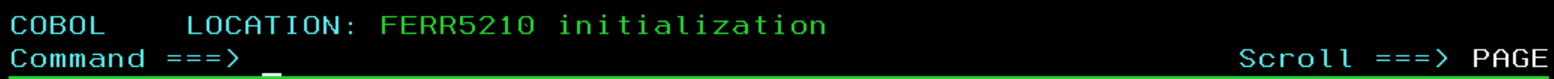


Fonte do programa exibido na janela Source.



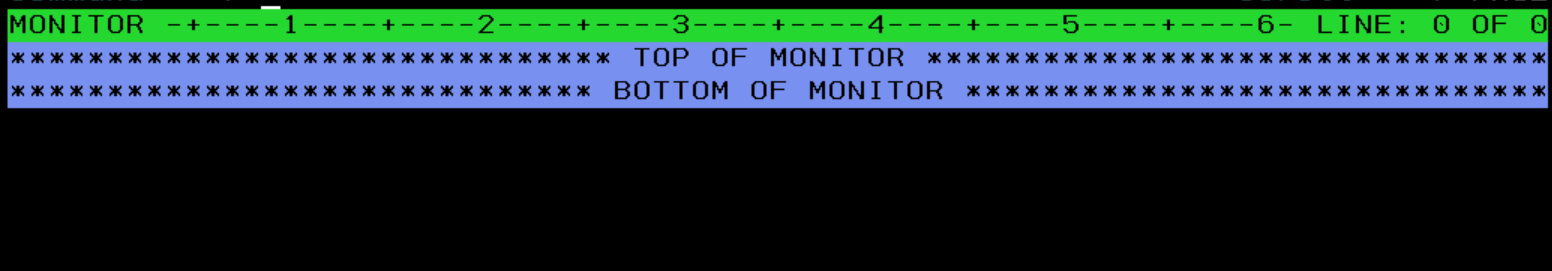
**Área de cabeçalho:**

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.



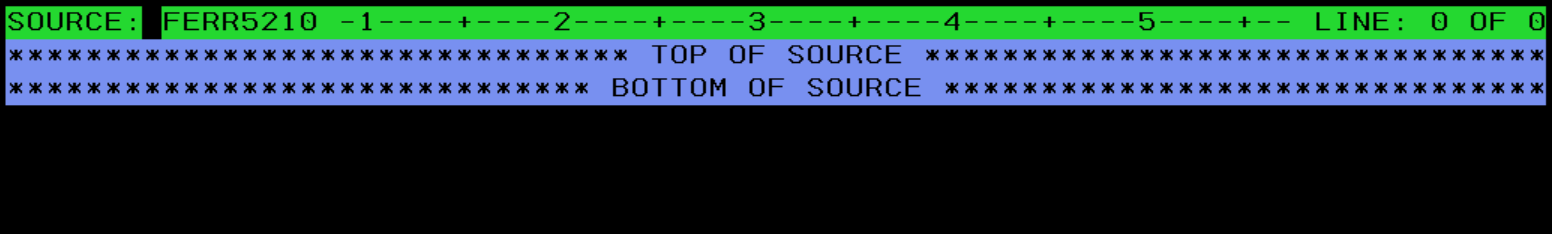
**Janela Monitor:**

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



**Janela Source (Fonte):**

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



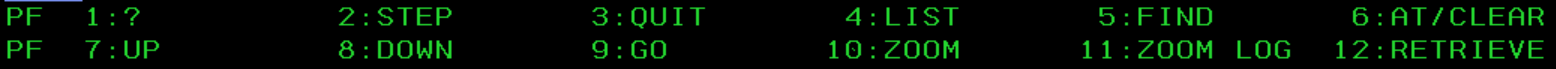
**Janela Log:**

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.



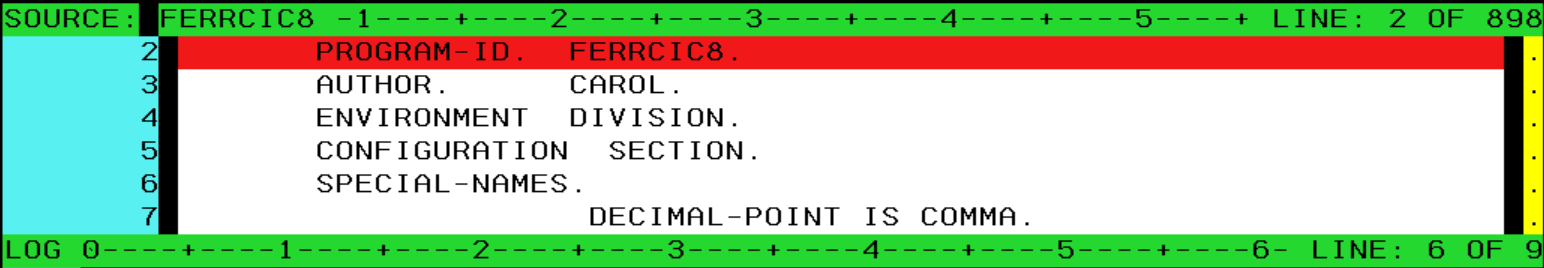
Mais adiante descreveremos as funções da PF´s e alguns outros comandos.

# **Descrição das PF´s.**

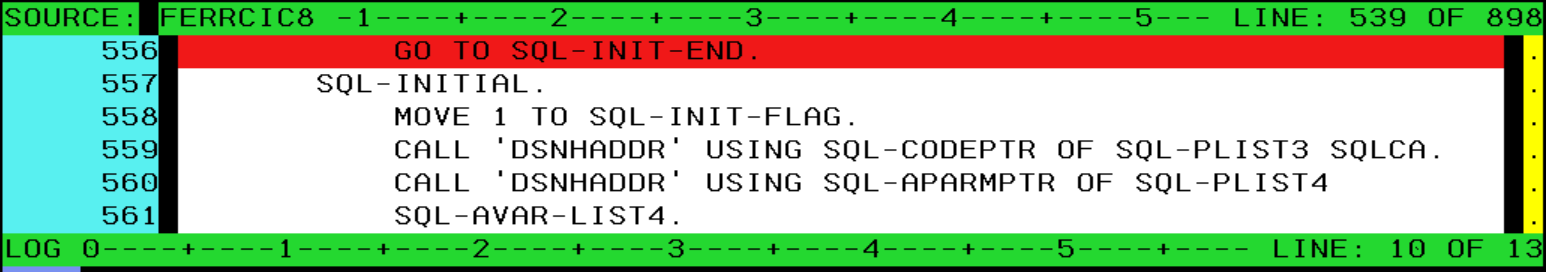
**PF2-STEP:**

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.

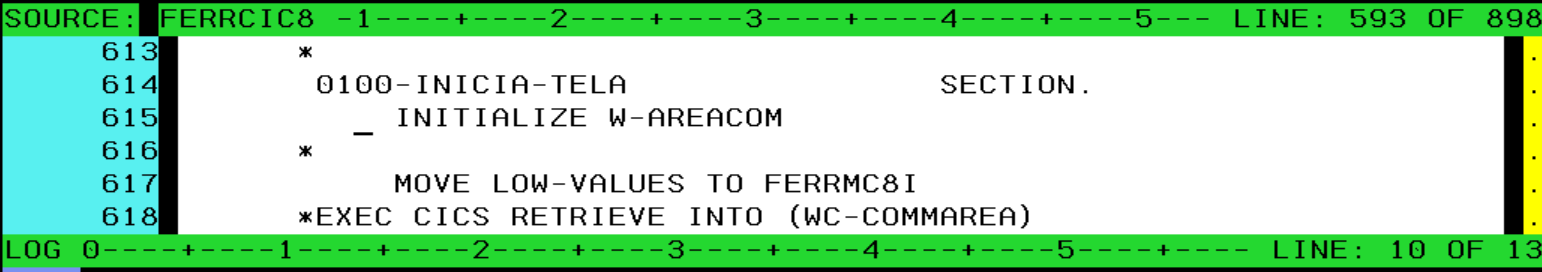


Após acionar <PF2> 3 vezes o programa é posicionado na primeira instrução da PROCEDURE DIVISION.

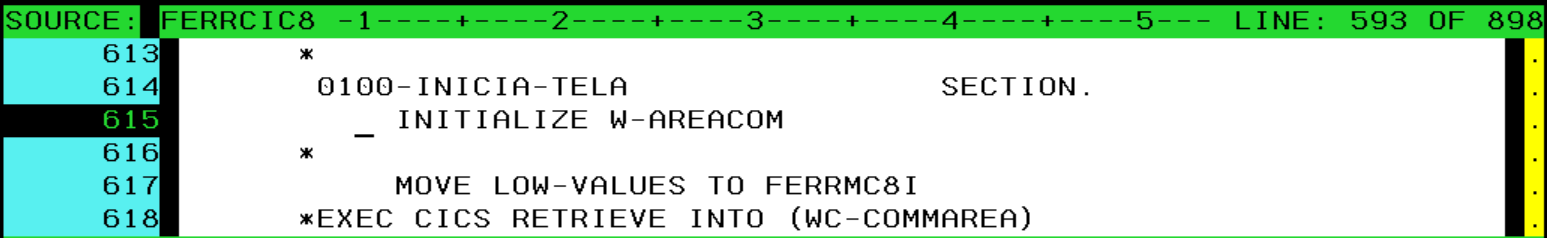


**PF6-AT/CLEAR:**

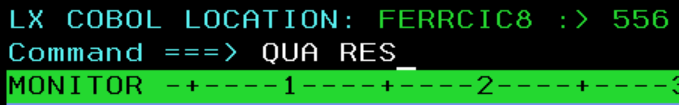
Insere ou retira um Break Point na linha onde o cursor estiver posicionado.

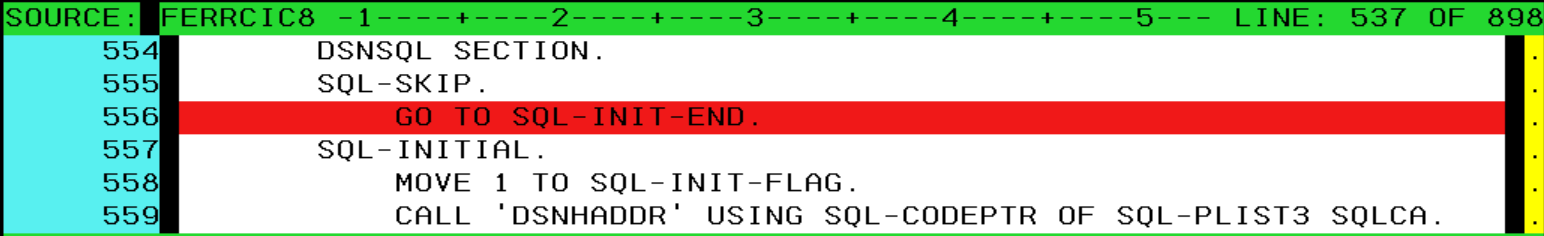


Após <PF6>, note que a linha 615 onde o curso estava ficou em “destaque” indicando um BREAK POINT.



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

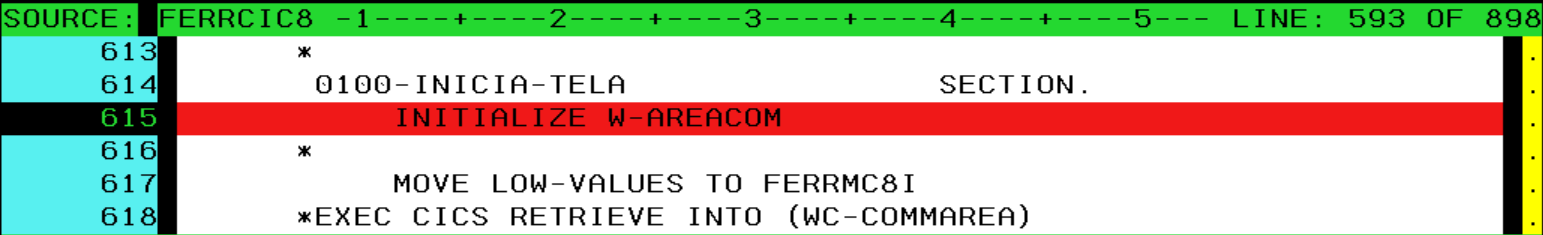




**PF9-GO:**

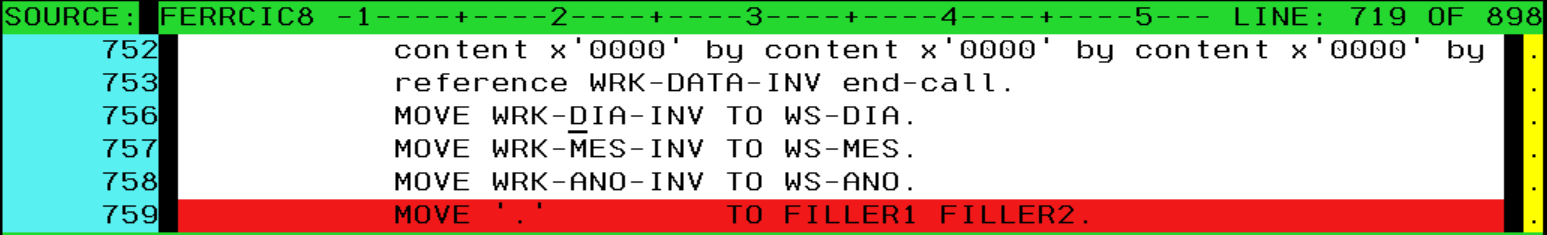
Executa o programa até o próximo BREAK POINT ou até o final caso não encontre BREAK POINT.

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

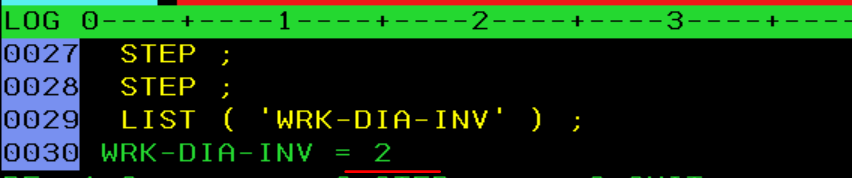


**PF4-LIST:**

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.



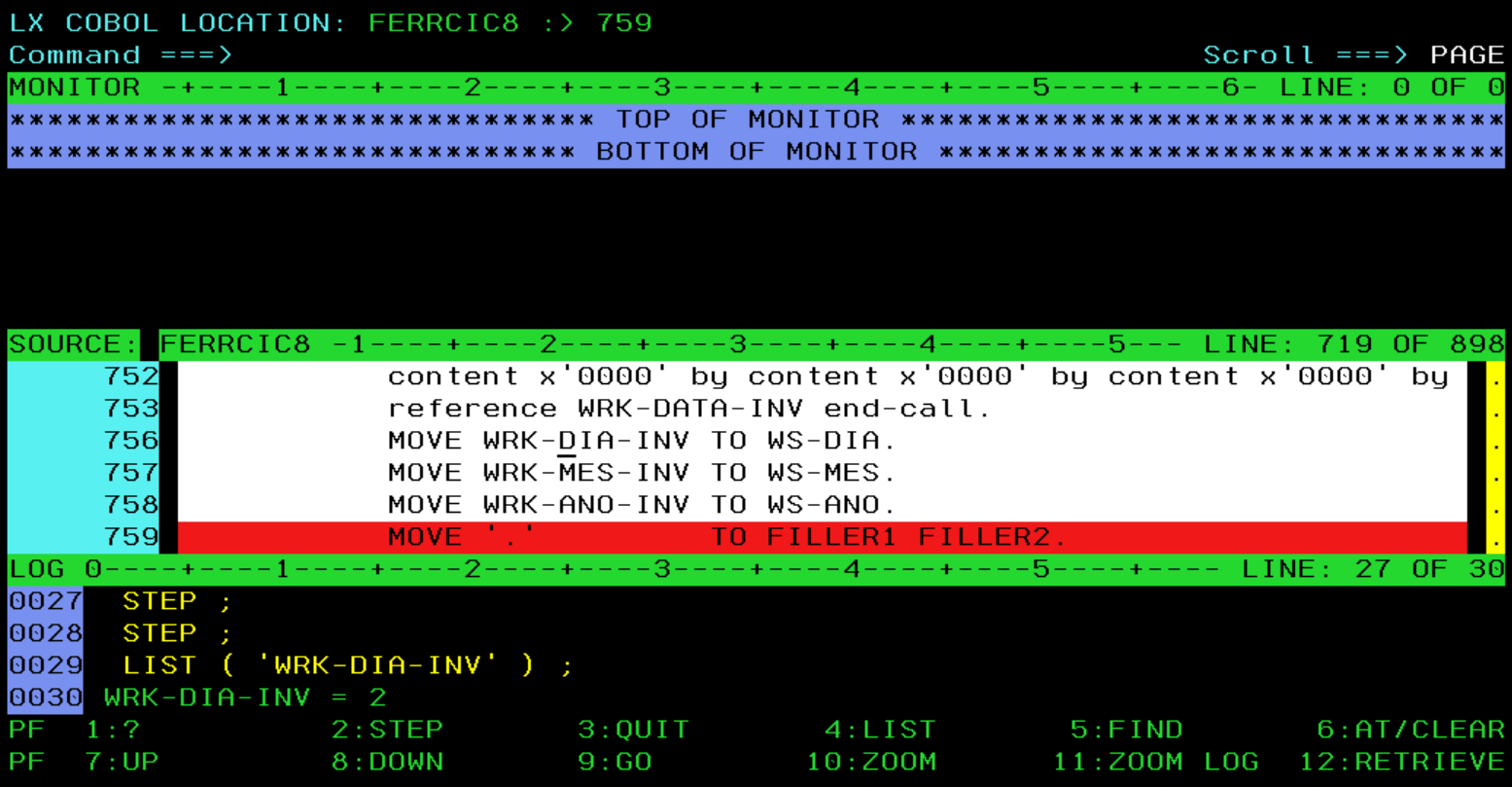
**PF11-ZOOM LOG:**

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

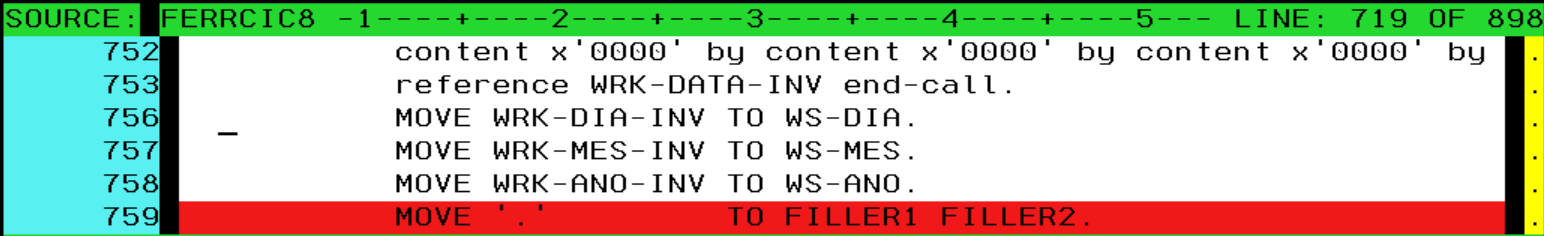


Outra <PF11> volta a janela ao normal.

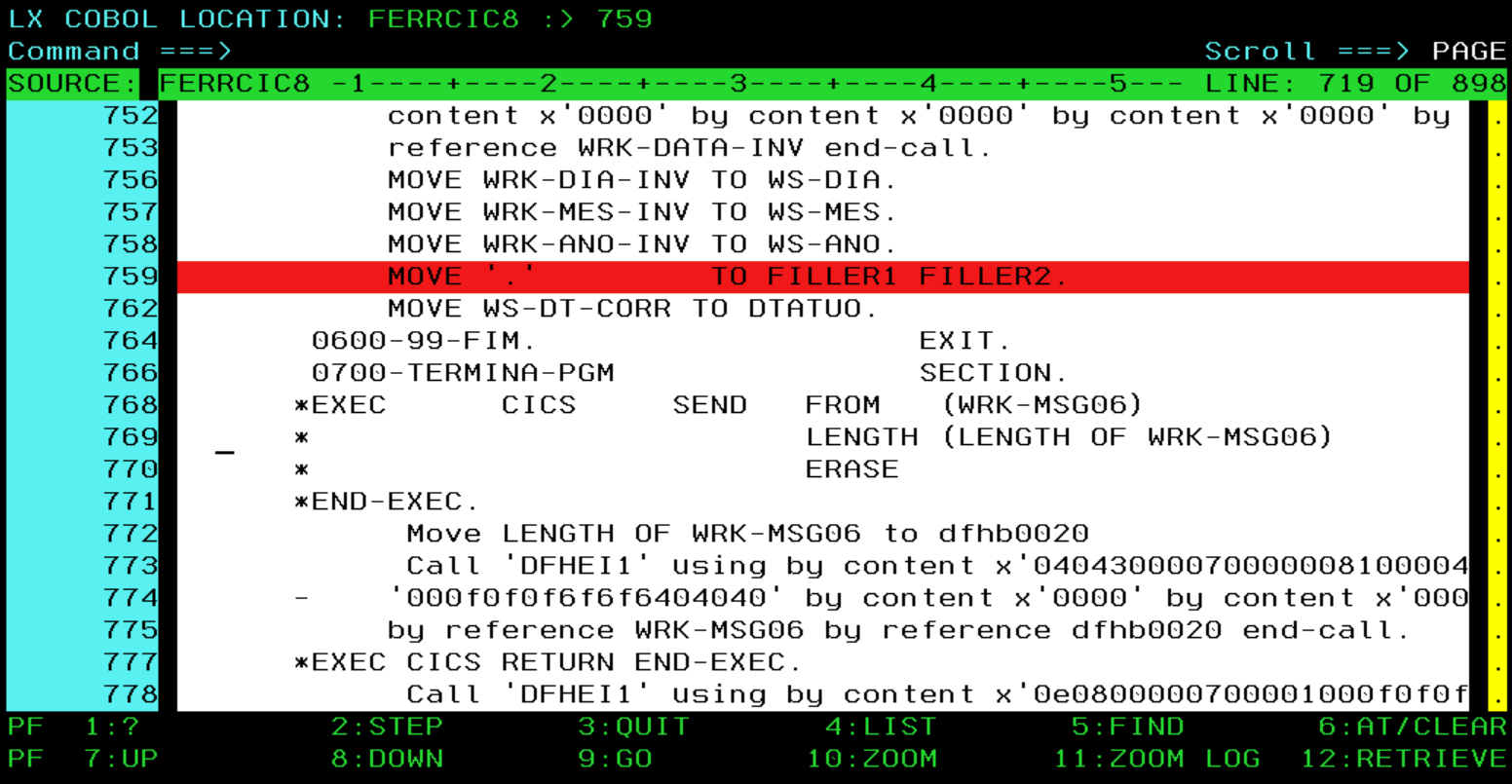


**PF10-ZOOM:**

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.

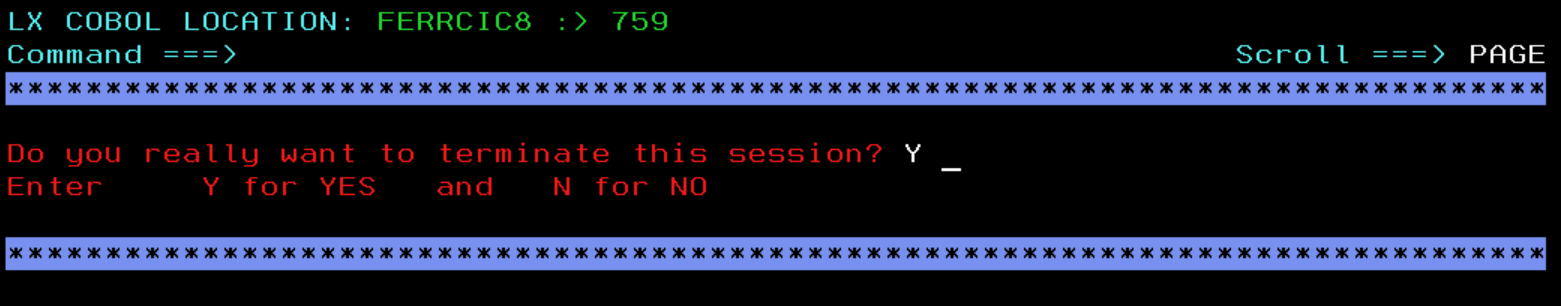


<PF10>, expande a janela



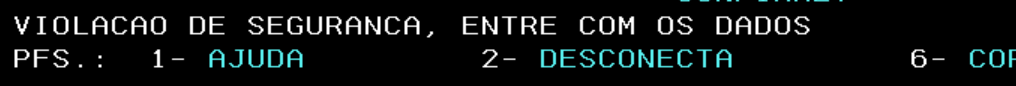
**PF3-QUIT:**

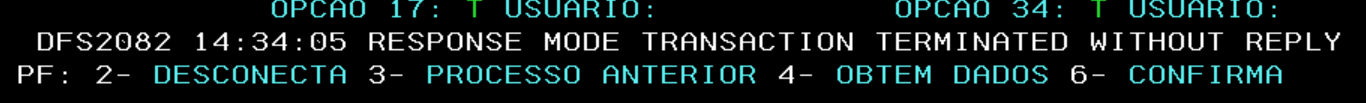
Encerra a sessão de Debug.



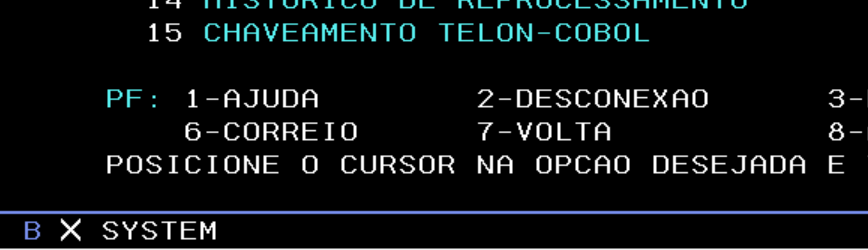
Tome cuidado quando utilizar o <Pf3>-Quit, pois o programa é interrompido no ponto em que estava “parado”, ele não executa até o final, assim dependendo do local que o programa é terminado, podemos ter resultados inesperados.

Seguem algumas mensagens recebidas na utilização da <Pf3>-Quit:





A transação simplesmente ficou em “X SYSTEM” e não retorna.



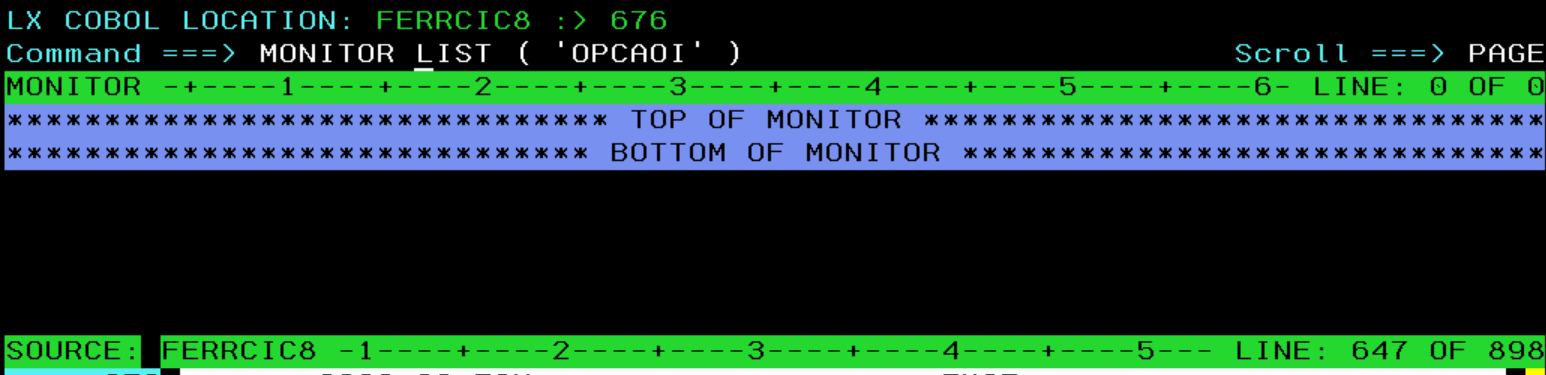
Normalmente com <Clear> ou <PA1> o controle retorna ao terminal.

**Observação:** após encerrar as atividades de Debug da transação IMS é muito importante fazer a “limpeza” dos recursos alocados, evitando travamento das suas transações e liberando recursos para outros usuário, utilize as instruções descritas no item [Encerrando o Debug de uma transação IMS.](#_Encerrando_o_Debug)

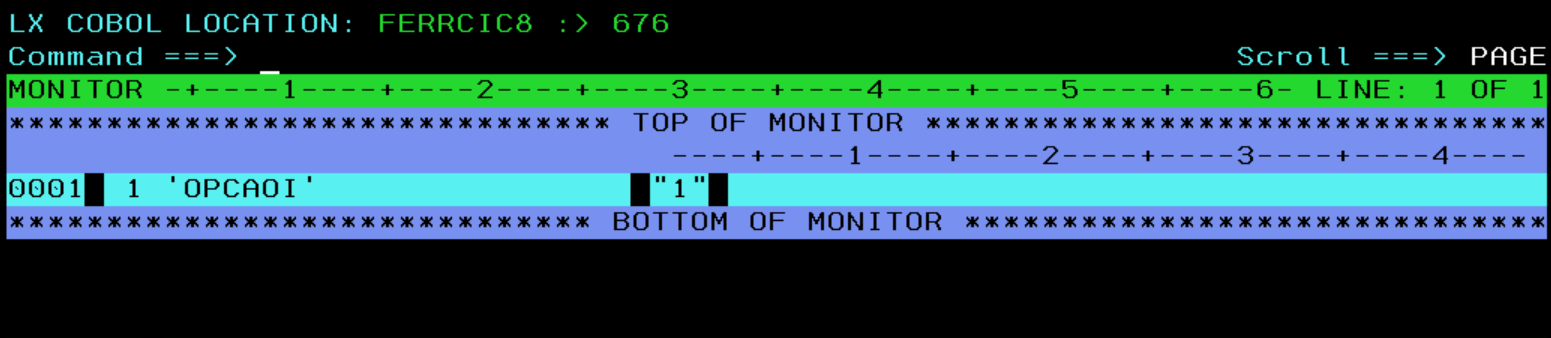
**Monitorando variáveis**

Modo I: Monitorando variáveis individualmente

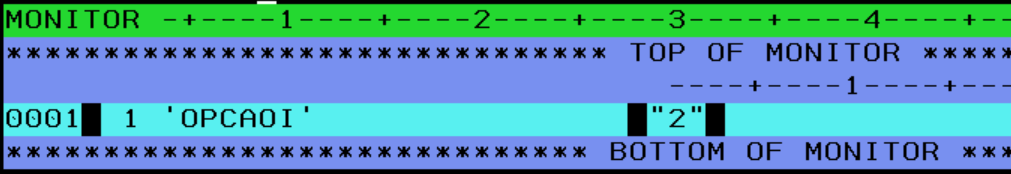
Entre com o comando MONITOR LIST ('nome-da-var')



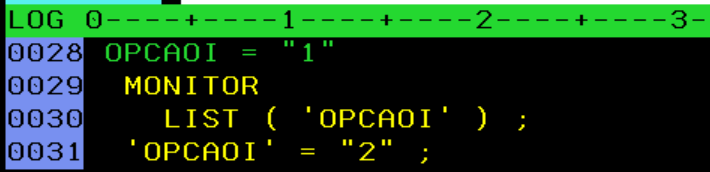
Será exibido o conteúdo da variável na janela Monitor, esta janela ficará fixa na tela sendo atualizada quando a variável sofrer atualização.



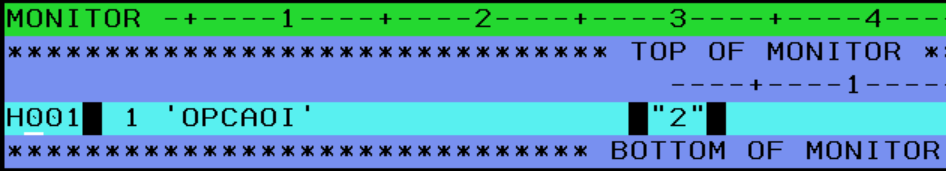
Também podemos alterar o valor da variável, basta digitar o novo valor na janela e teclar <Enter> para efetivar.



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>.





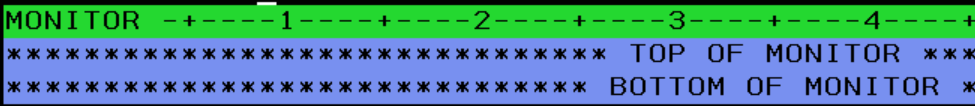
Para voltar ao modo “Default”, digite “D” no número em frente ao nome da variável e <Enter>.



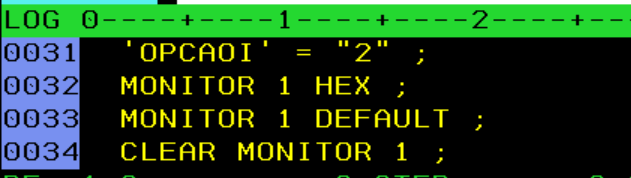


Digite “C” para fechar (“limpar”) o monitor de uma variável.





Note que na janela de Log aparecem os comandos correspondentes:



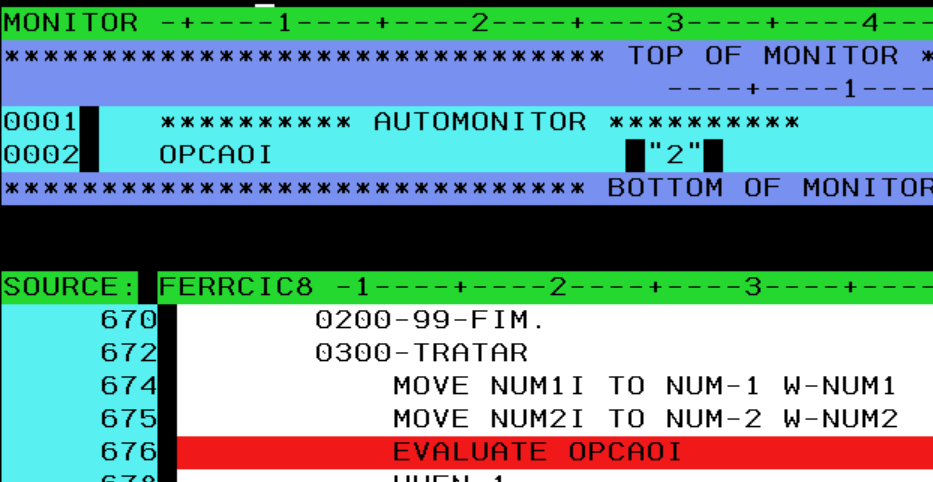
Modo II: Monitorando variáveis automaticamente

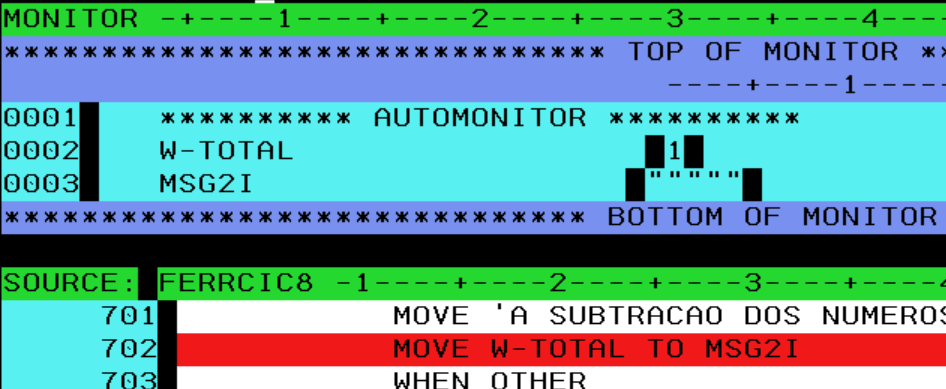
Neste caso, serão monitoradas automaticamente as variáveis envolvidas no comando executado.

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.



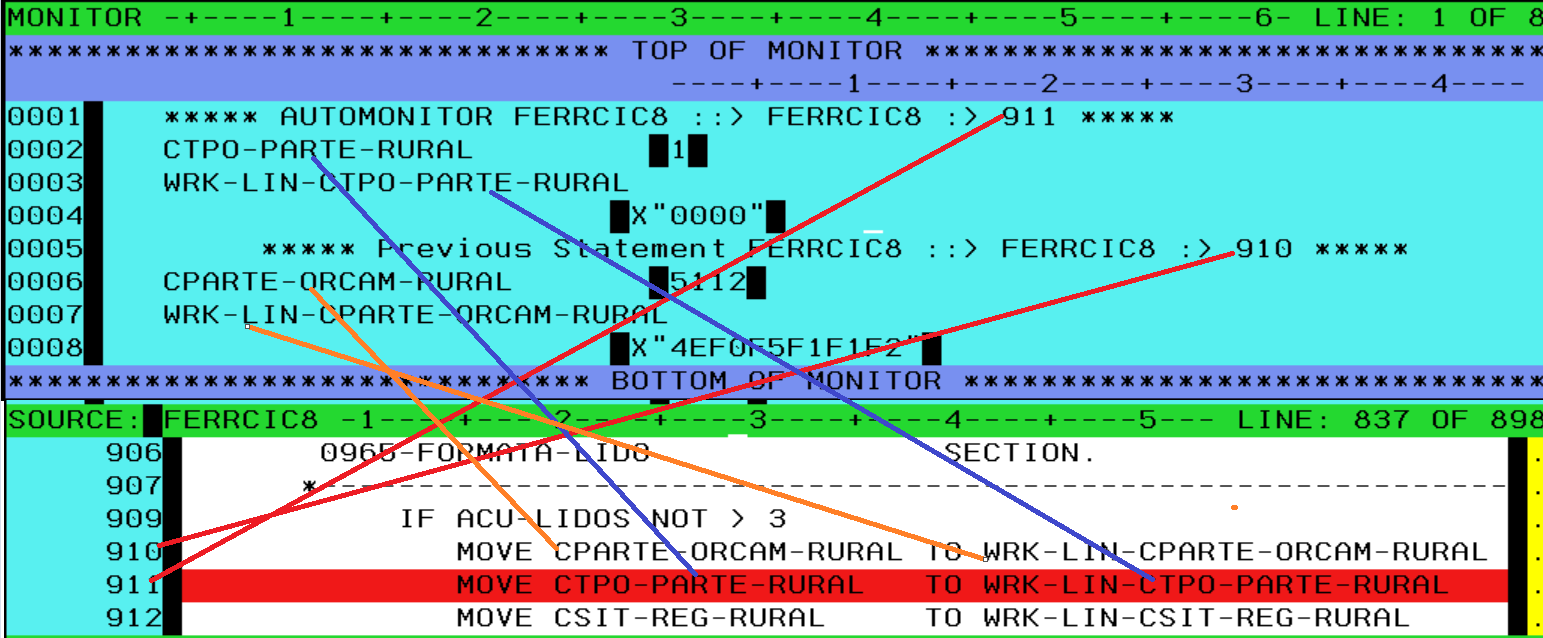


Entre com o comando SET AUTO ON BOTH<Enter>.

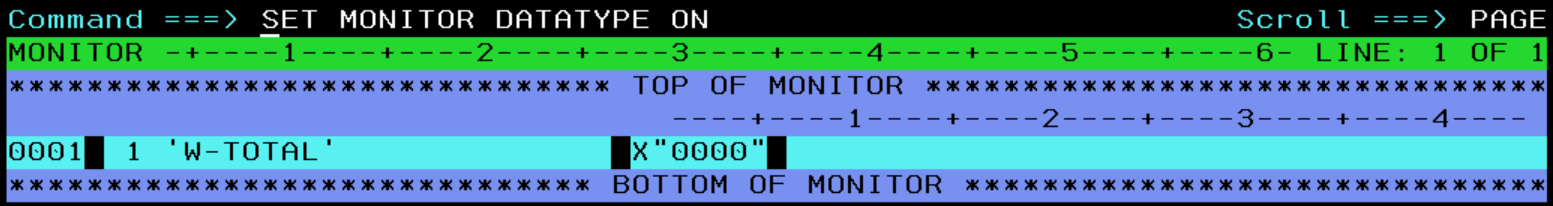
De acordo com as variáveis envolvidas no comando atual e anterior a janela de monitor vai mudando, mostra a variável do comando anterior e o atual antes da execução do comando.

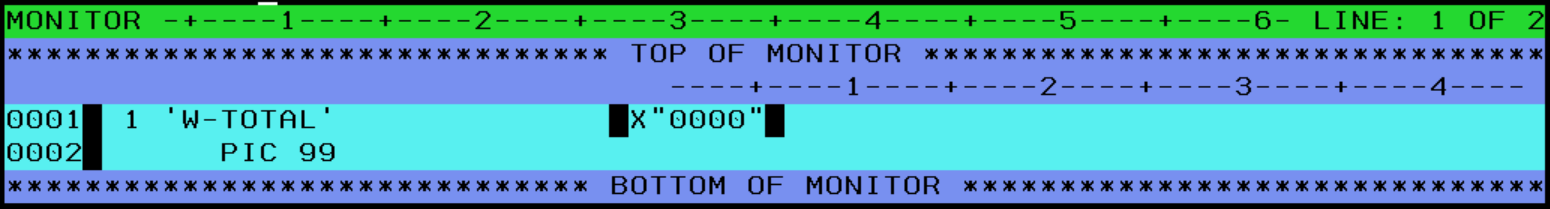


Mostra na parte superior da janela Monitor as variáveis da instrução corrente (linha 911) e na parte inferior as variáveis da instrução anterior (linha 910).



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>.

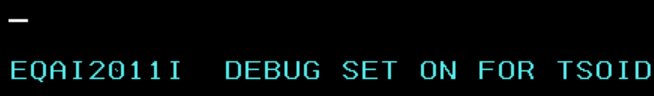
# **Encerrando o Debug de uma transação IMS.**

Para liberar os recursos utilizados no Debug, execute os passos descritos a seguir.

Encerre a execução do seu programa e tecle <Clear>.

**Liberar usuário no IMS:** Assinale o TSOID=OFF na transação EQASET:





Dê <Clear> e saia do IMS (/RCL).



**Encerre a sessão do TIM:** utilize a <Pf3>(EXIT).

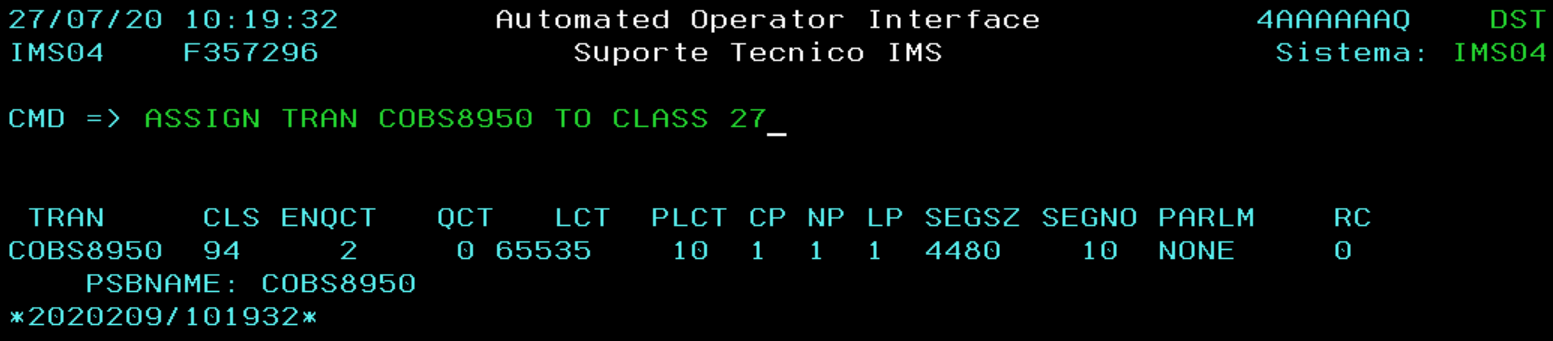
**Desligar no “arquivo de configuração”:** Entre no TS48, opção 190 (Debug Tool), na opção 6 novamente e altere o campo “Test Option” de TEST para NOTEST.



Digite <Enter> para confirmar e <Pf3> (3 vezes) para sair do Debug Tool e mais uma <Pf3> para sair do sair do TSO.

**Retornar transação para a classe original:** Entre no IMS, faça SIGNON (<Pf6>), e entre na transação STIDST e retorne sua transação para a classe original (nos dois IMSs, IMS04 e IMS06).

Utilize o comando: “ASSIGN TRAN tttttttt TO CLASS 27”, ONDE: tttttttt é o código da transação.



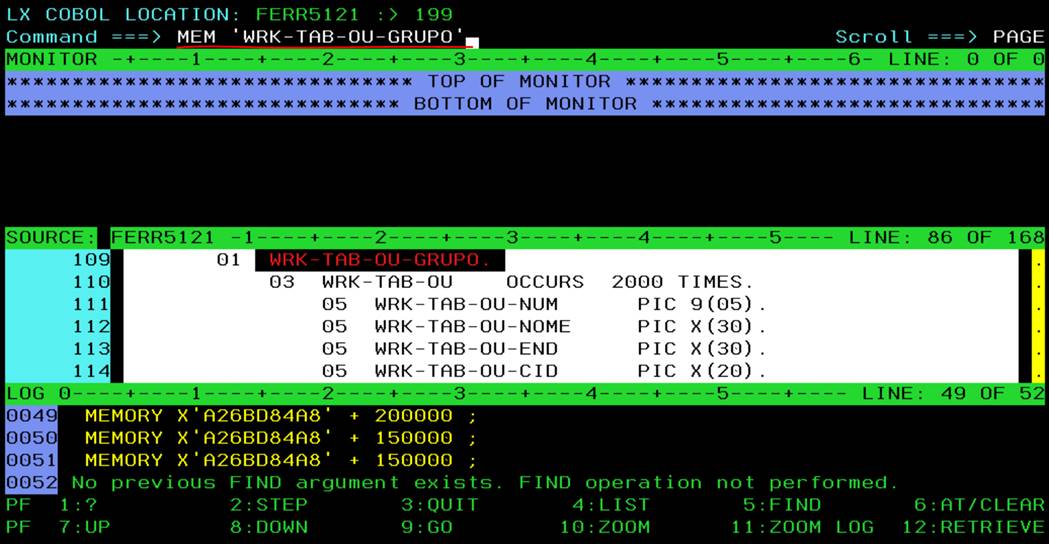
Dê <Clear>, seguido /RCL para sair do IMS.



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

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

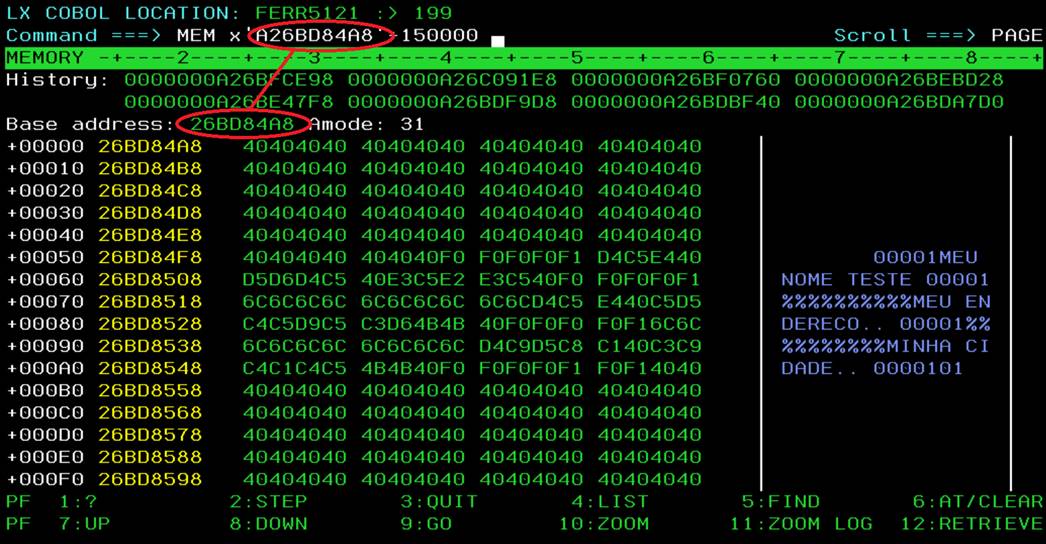
**Comando: MEM 'WRK-TAB-OU-GRUPO'**



**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:

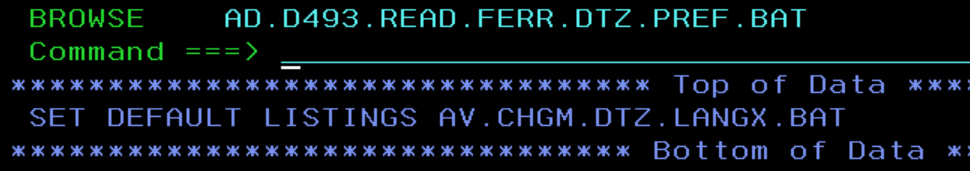
Organization . . . : PS

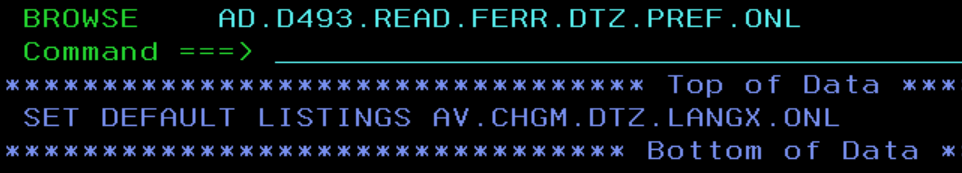
Record format . . . : FB

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>