**Sumário**

[**1.** **Preparando programa para Debug no Changeman.** 2](#_Toc47450963)

[**2.** **Preparando arquivo de configuração para a primeira execução.** 4](#_Toc47450964)

[**3.** **Acessando o DTZ (Debug Tool)** 7](#_Toc47450965)

[**4.** **Capturando um JCL para execução do programa** 10](#_Toc47450966)

[**5.** **Informando parâmetros para a execução - Foreground.** 14](#_Toc47450967)

[**6.** **Executando um programa com o Debug Tool - Foreground.** 16](#_Toc47450968)

[**7.** **Painel padrão do Debug Tool.** 17](#_Toc47450969)

[**8.** **Carregando a listagem de um programa.** 19](#_Toc47450970)

[**9.** **Informando parâmetros para a execução - Background.** 20](#_Toc47450971)

[**10.** **Fazendo Logon no TIM-Terminal Interface Manager.** 22](#_Toc47450972)

[**11.** **Executando um programa com o Debug Tool - Background.** 23](#_Toc47450973)

[**12.** **Descrição das PF´s.** 25](#_Toc47450974)

[**13.** **Monitorando variáveis.** 30](#_Toc47450975)

[**14.** **Lista de comandos e PF´s.** 34](#_Toc47450976)

[**15.** **Informando parâmetros para a execução – Foreground – Assembler.** 37](#_Toc47450977)

[**16.** **Executando um programa com o Debug Tool – Foreground - Assembler.** 39](#_Toc47450978)

[**17.** **Informando parâmetros para a execução – Background - Assembler.** 41](#_Toc47450979)

[**18.** **Executando um programa com o Debug Tool – Background - Assembler.** 43](#_Toc47450980)

[**19.** **Apêndice I – Inspecionando variáveis na memória.** 44](#_Toc47450981)

[**20.** **Apêndice II – Passando parâmetros via PARM.** 46](#_Toc47450982)

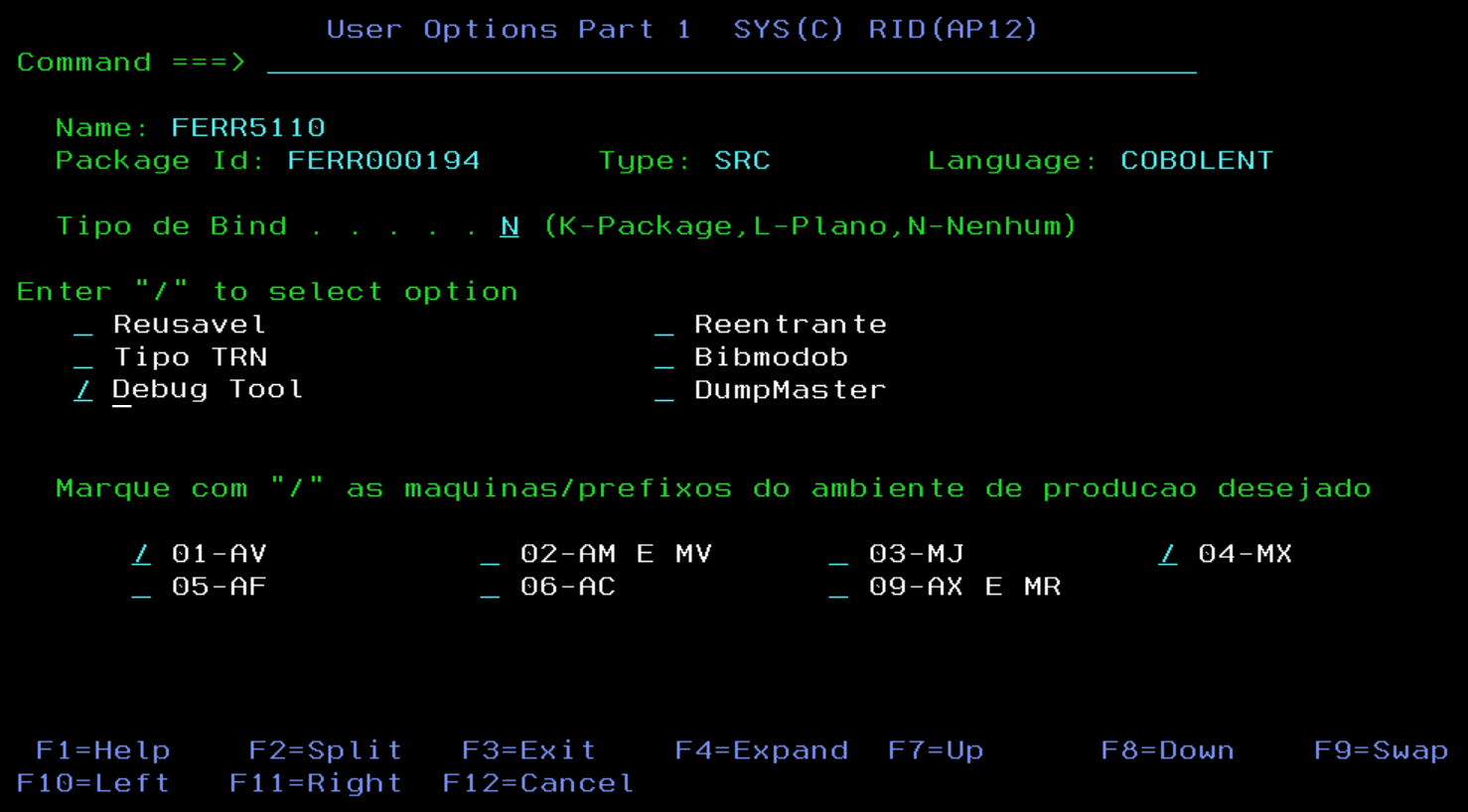
[**21.** **Apêndice III – Arquivos de configuração** 47](#_Toc47450983)

[**22.** **Apêndice IV – Link para manuais da IBM.** 48](#_Toc47450984)

**IBM – Debug Tool - Batch**

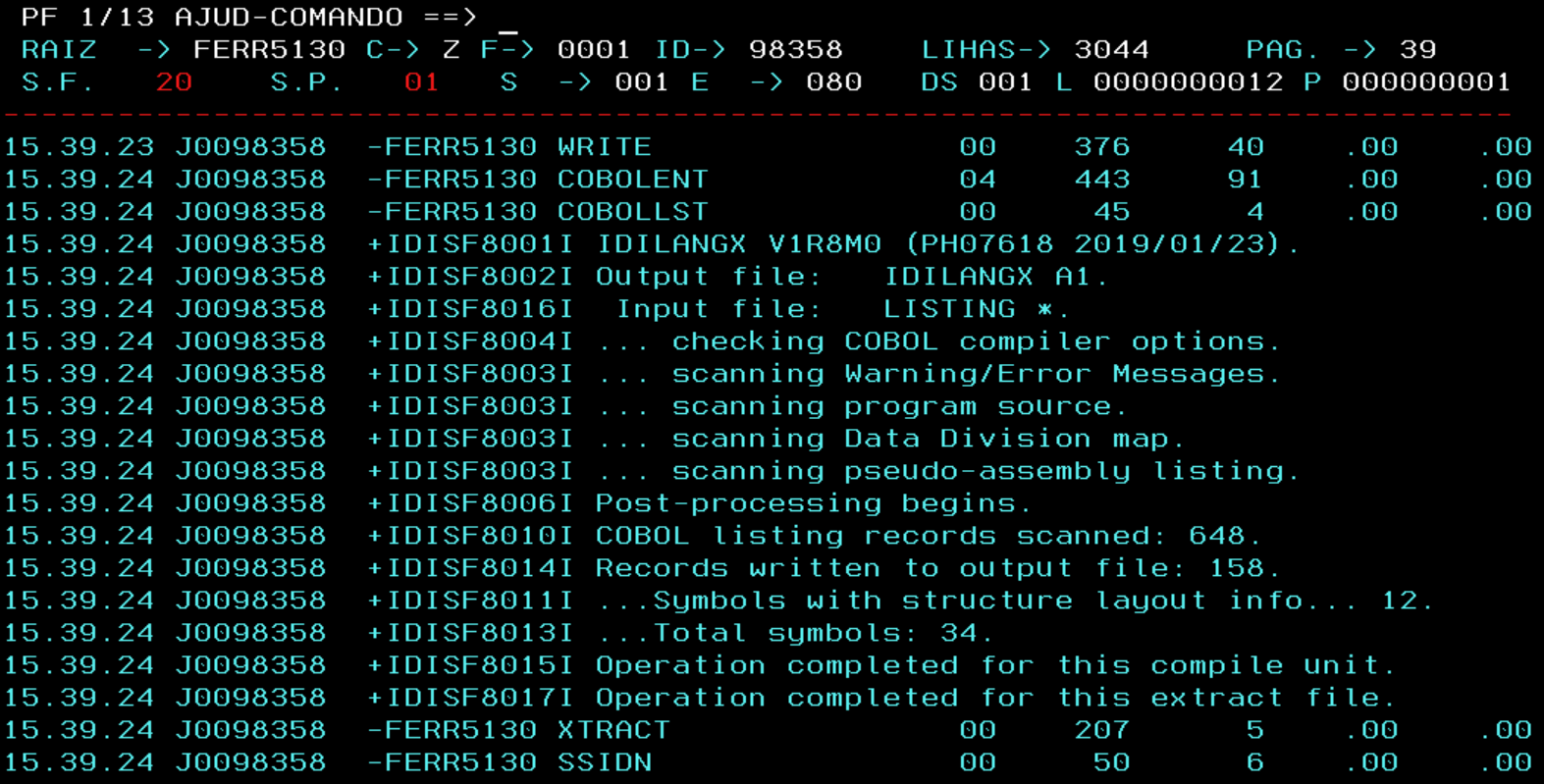
# **Preparando programa para Debug no Changeman.**

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.

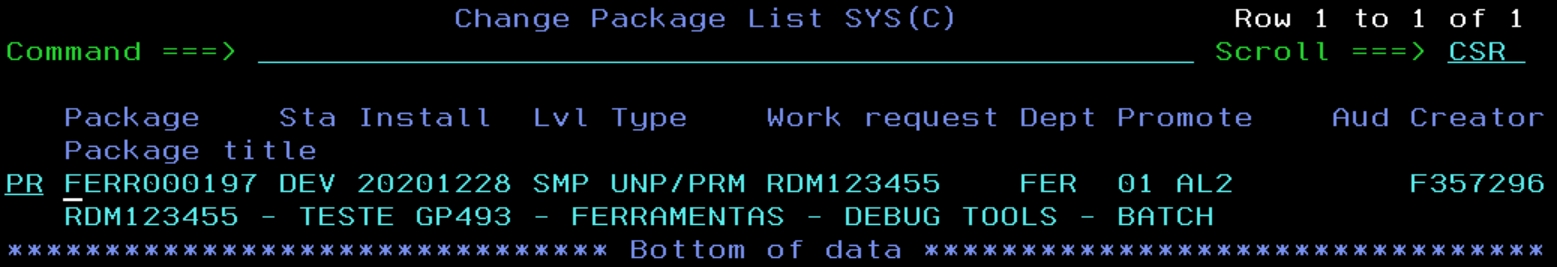


Na execução do JOB de compilação a listagem do programa será direcionada para o arquivo AV.CHGM.DTZ.LANGX.BAT que será utilizada no processo de Debug.

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

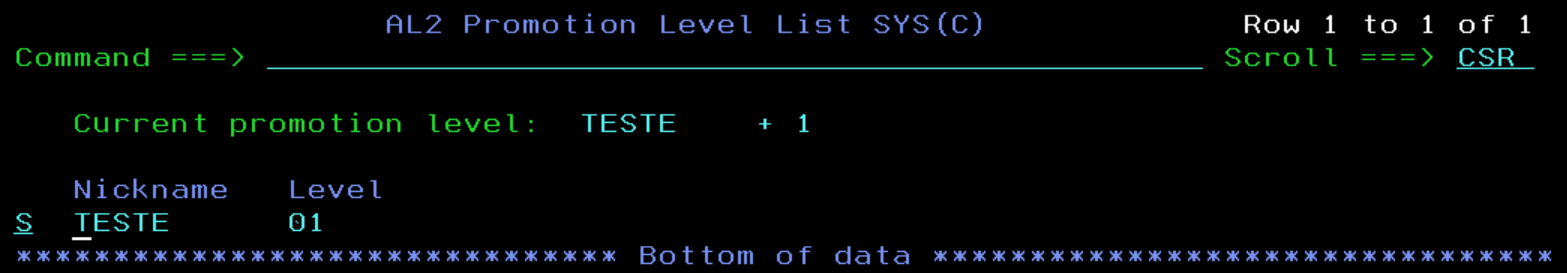


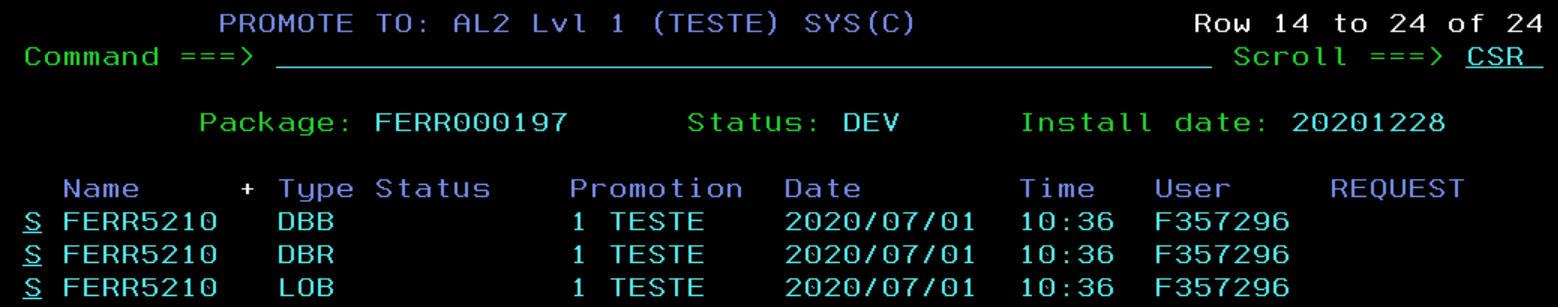
Após a compilação, devemos realizar o processo de **PROMOTE** para o ambiente de TU.









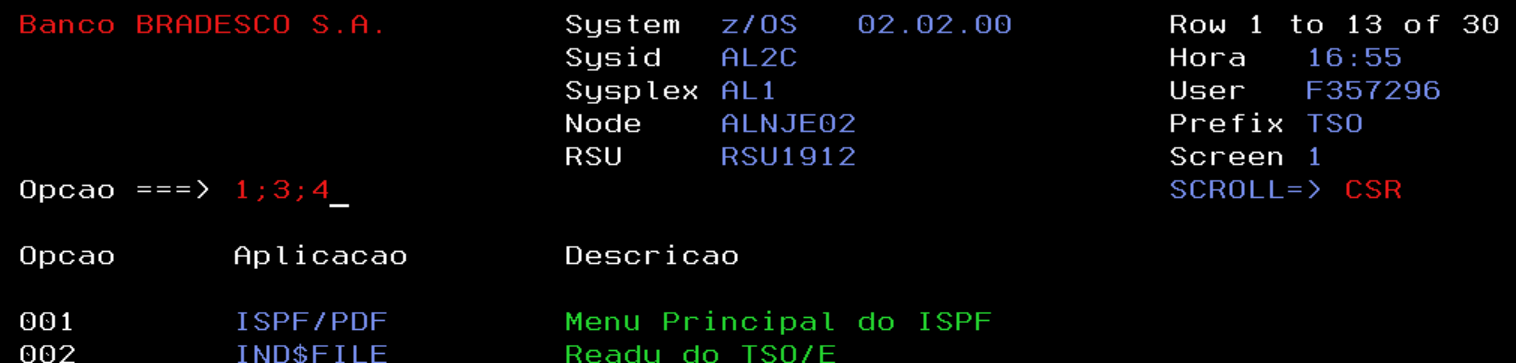


# **Preparando arquivo de configuração para a primeira execução.**

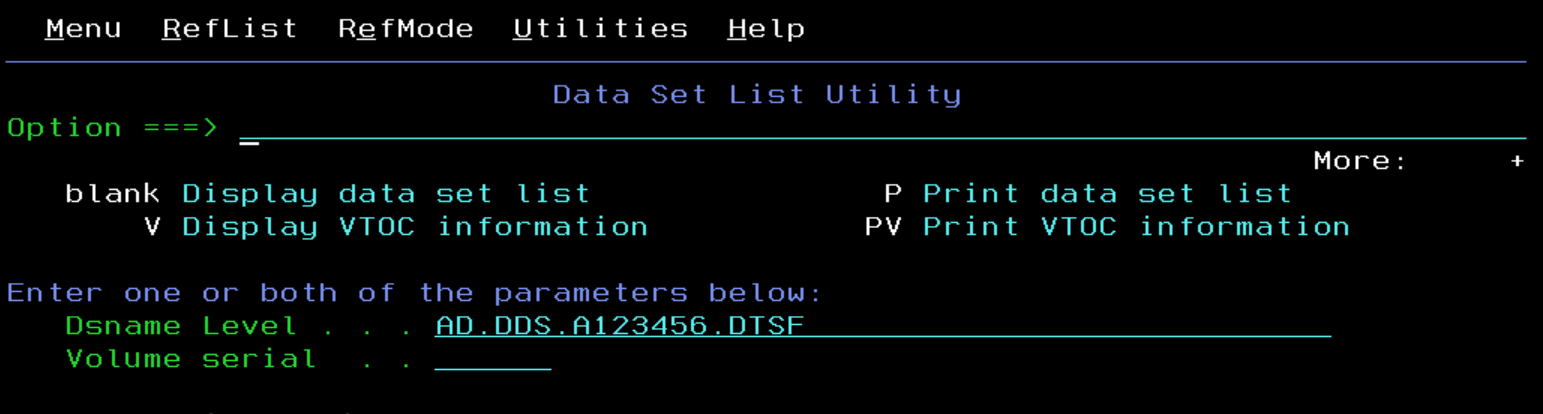
Para a utilização do IBM-Debug Tool for z/Os (ou abreviadamente DTZ) para programas BATCH é necessária a alocação de um arquivo (somente na primeira utilização) de configuração, denominado DTSF (Debug Tool Setup File).

Faremos uma cópia de um arquivo “padrão” (AD.DDS.A123456.DTSF) para um arquivo particular do usuário adotando como padrão de DSName: AD.DDS.<chave do usuário>.DTSF, onde <chave do usuário> deve ser substituído pela sua chave.

Utilize a opção 1;3;4 para fazer a cópia:



Informar o arquivo origem: AD.DDS.A123456.DTSF.



Selecionar com “/”.



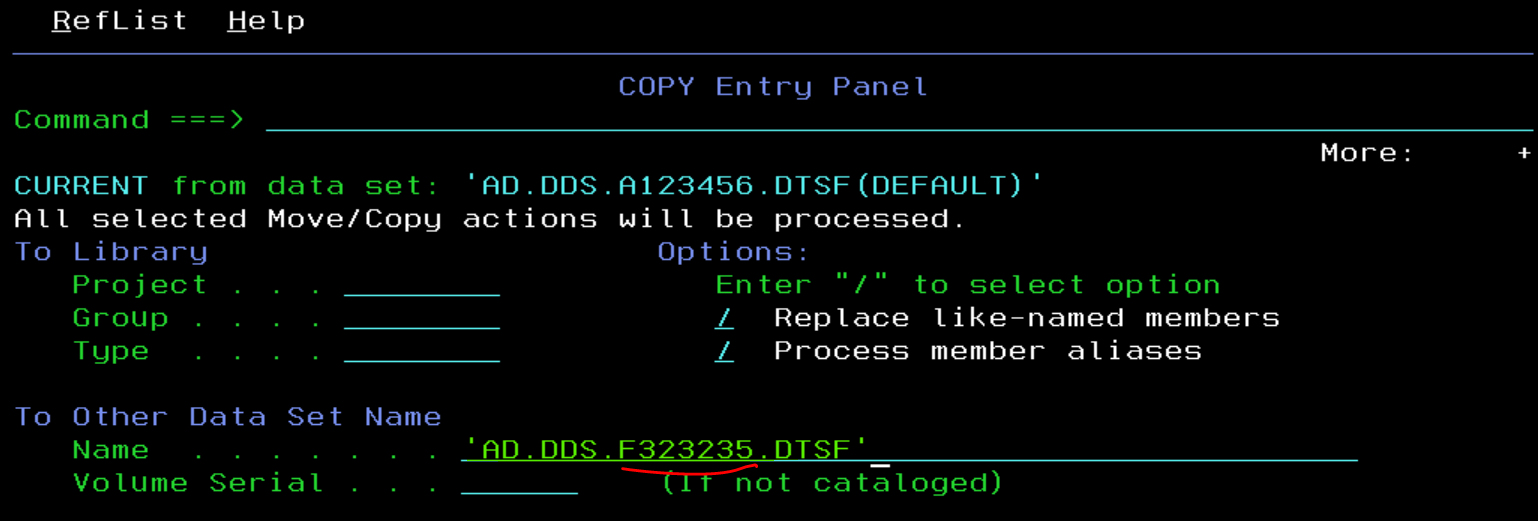
Escolher a opção 17. Copy.



Selecione todos, com “S \*” e tecle <Enter>.



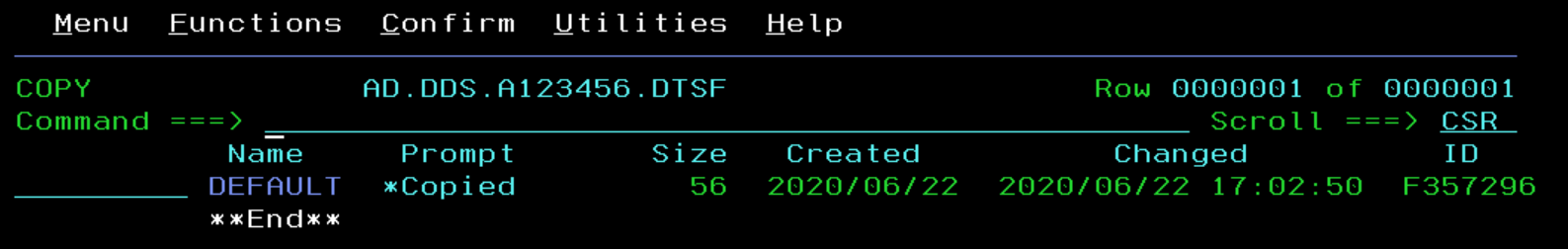
Informe o nome do arquivo no padrão: AD.DDS.<chave do usuário>.DTSF



Escolha a opção 1, para alocar o arquivo com as mesmas características do original.



Arquivo alocado e copiado com sucesso:

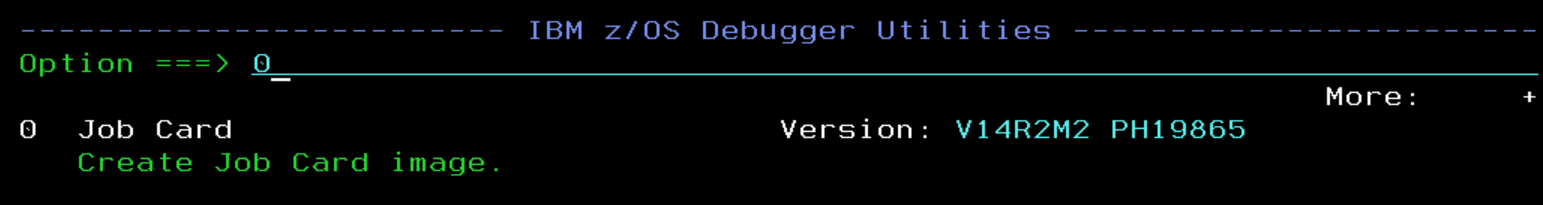


# **Acessando o DTZ (Debug Tool)**

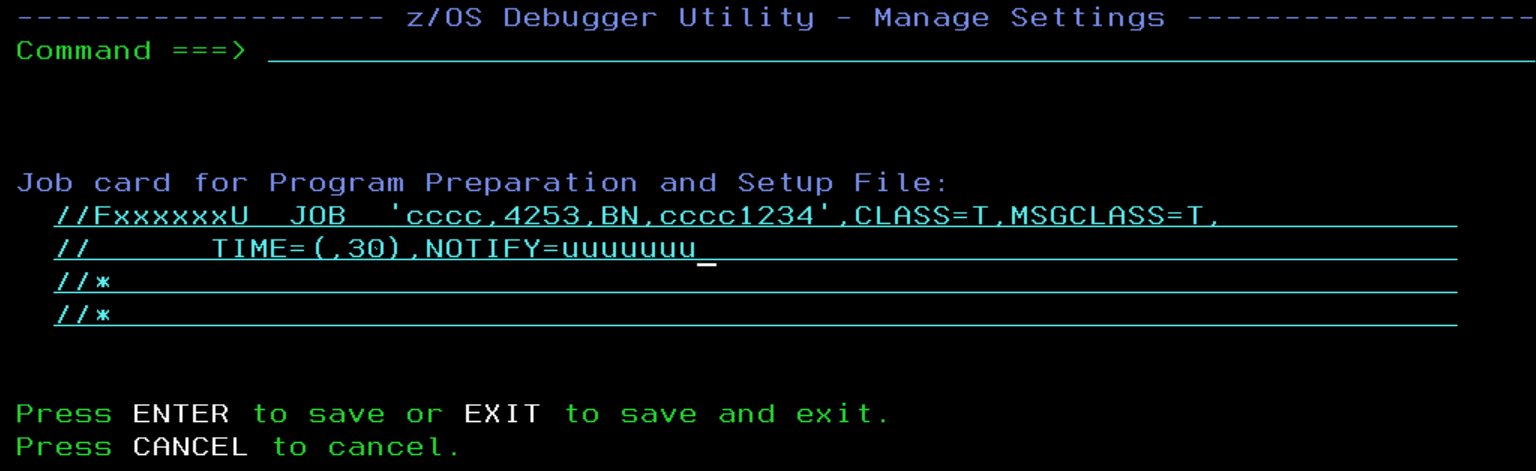
Para acesso a ferramenta, utiliza a opção 190 no TS48 (AL2C):



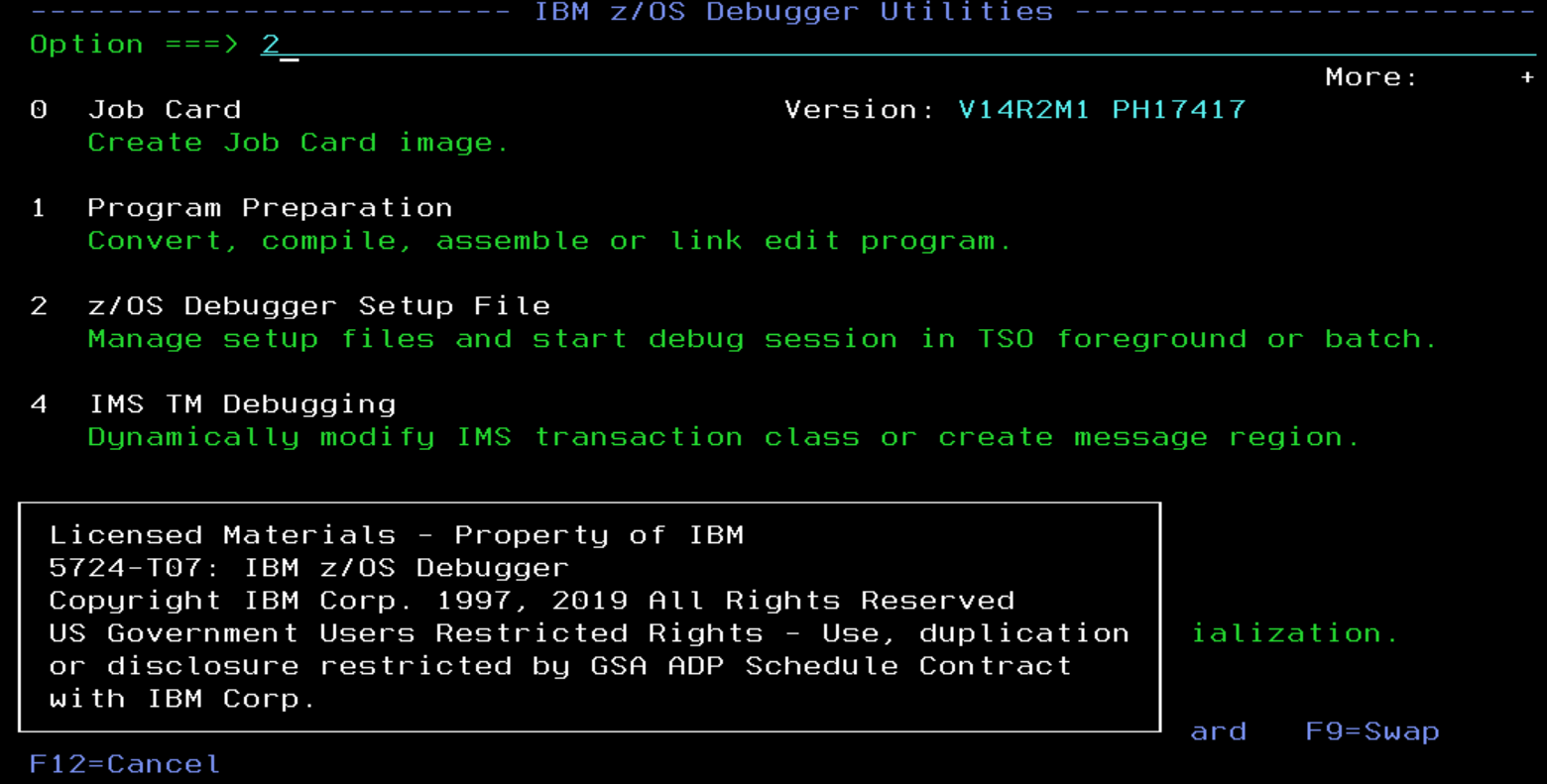
Utilize uma única vez a opção 0 (Job Card) para informar ao Debug Tool os de “JOB” que será utilizado.



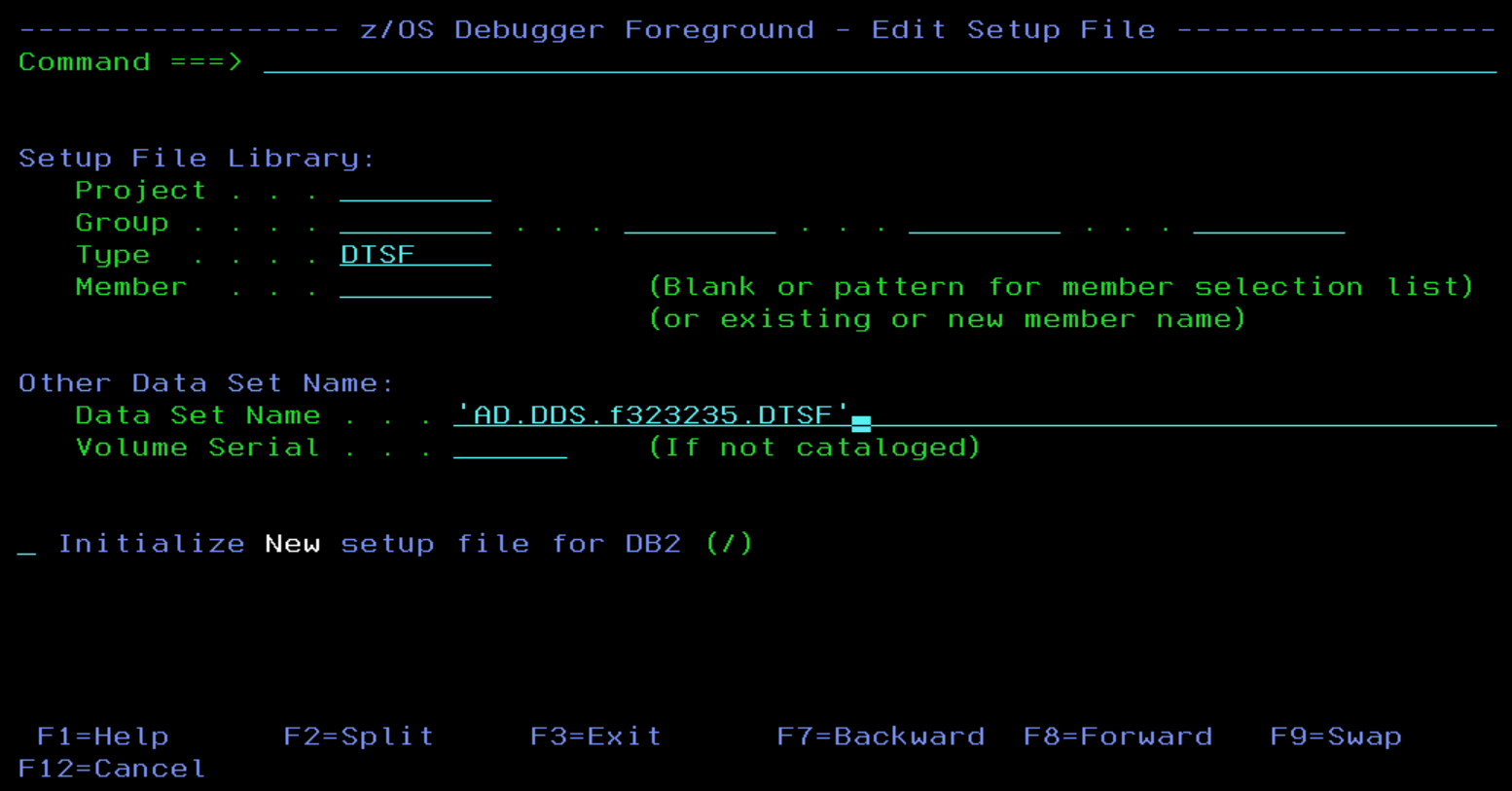
Informe o “cartão” de JOB que você utiliza nos seus JOB´s, tecle <Enter> para confirmar e em seguida <Pf3> para retornar ao menu.



Em seguida, utilize a opção 2.



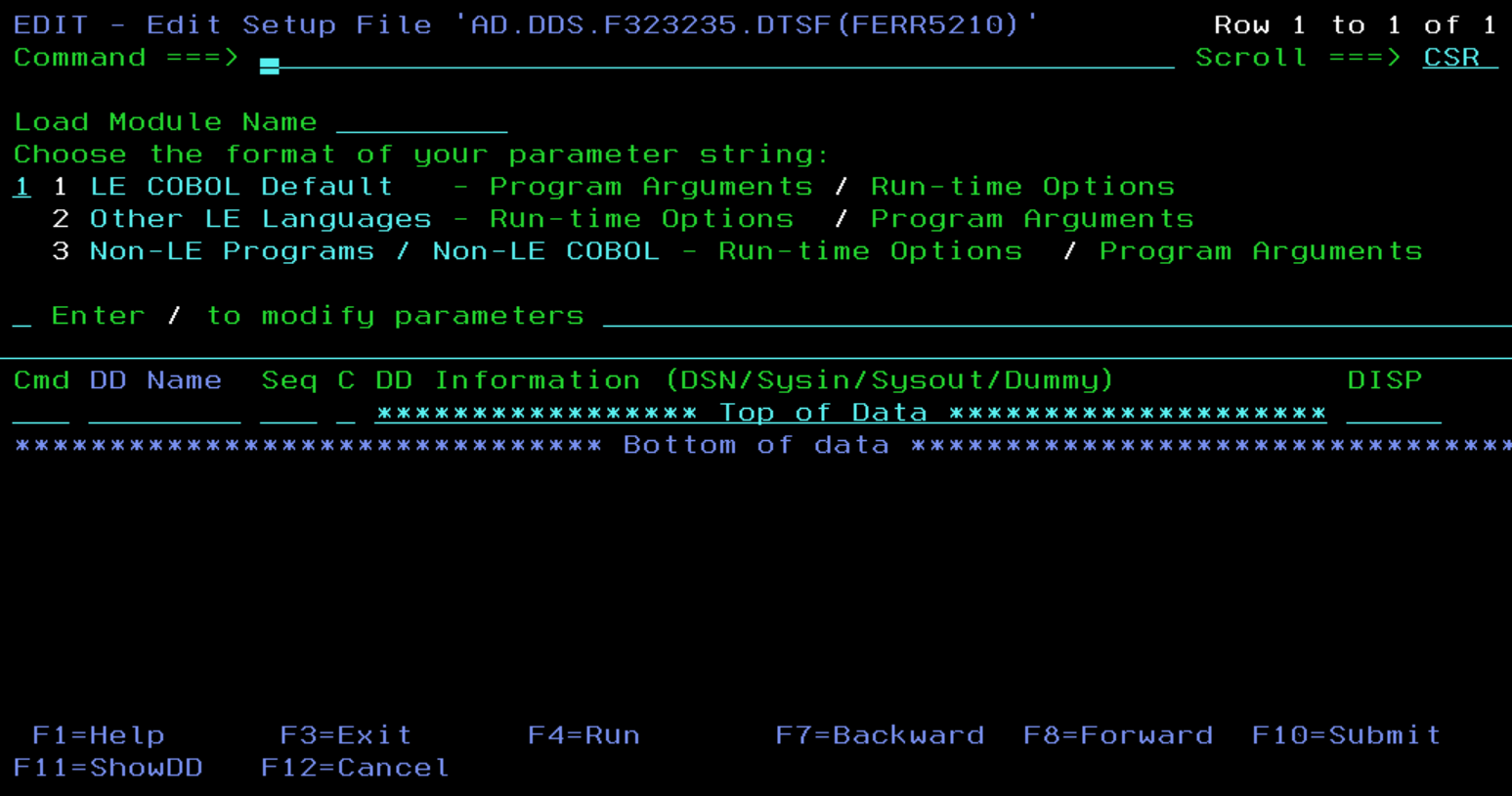
Informe o arquivo DTSF previamente alocado e tecle <Enter>



Se for sua primeira execução, digite na linha de comando “SELECT <código do programa>, senão selecione com “S” o programa desejado na lista.



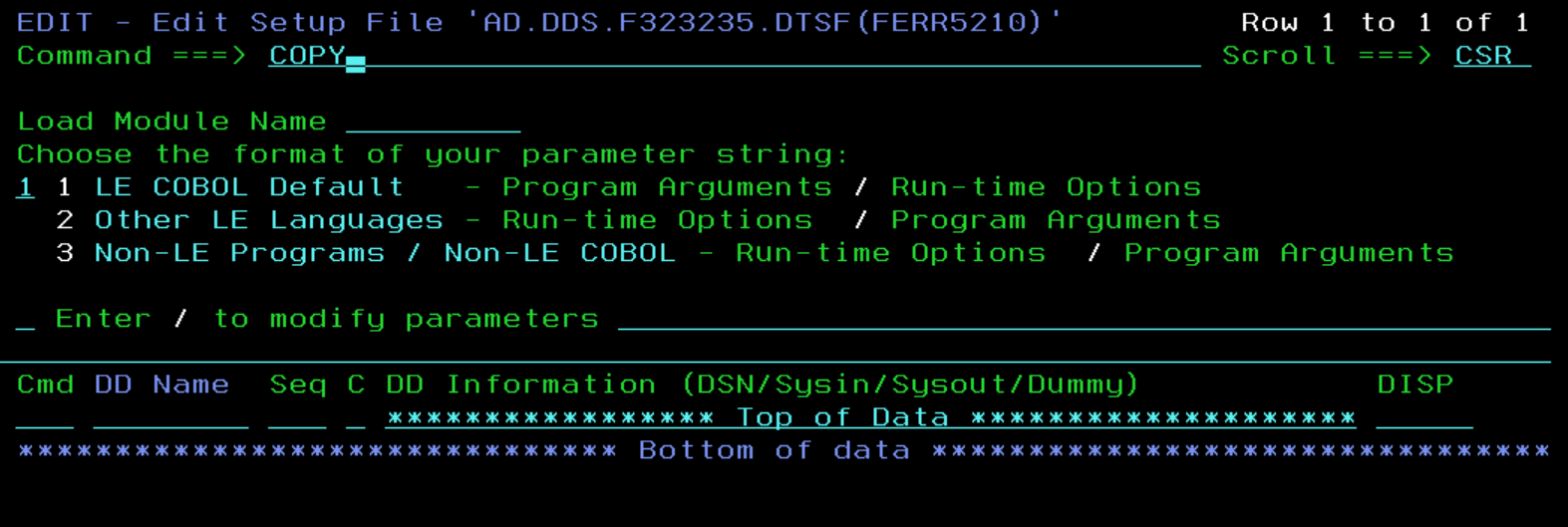
Será exibida a tela:



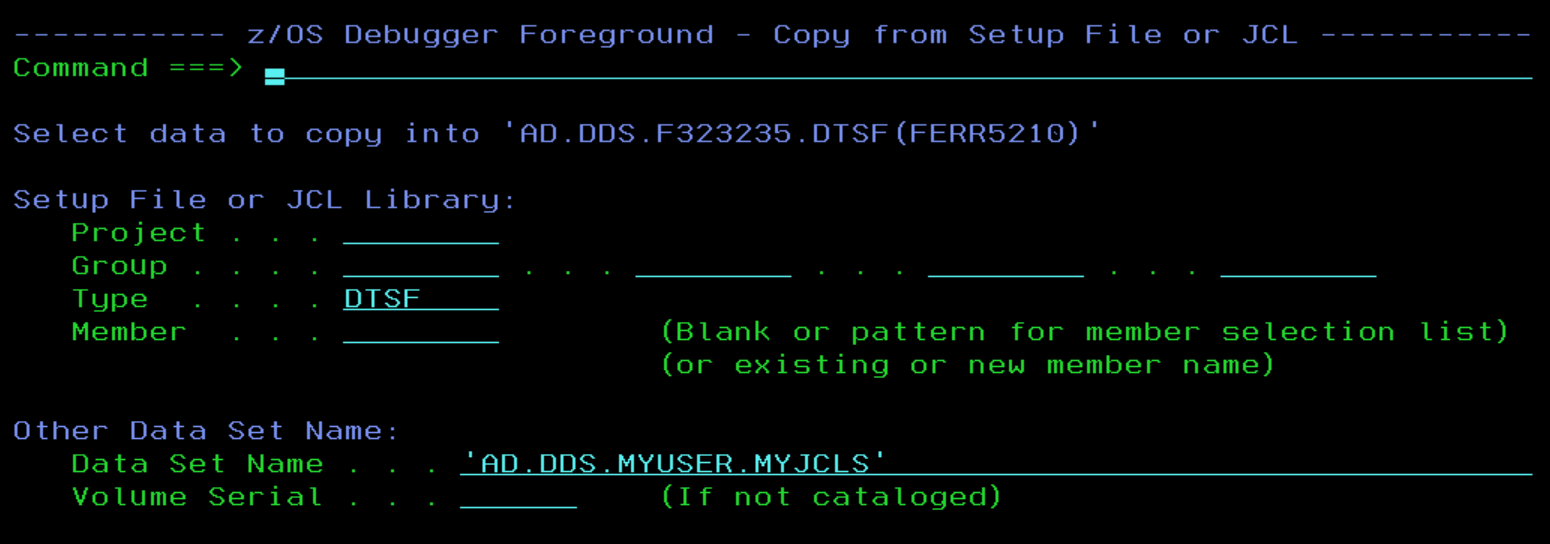
# **Capturando um JCL para execução do programa**

Para a execução do programa é necessário fazer a “captura” do JCL para execução sob o Debug Tool, para isso o JCL deve estar em um arquivo PDS ou PDSE.

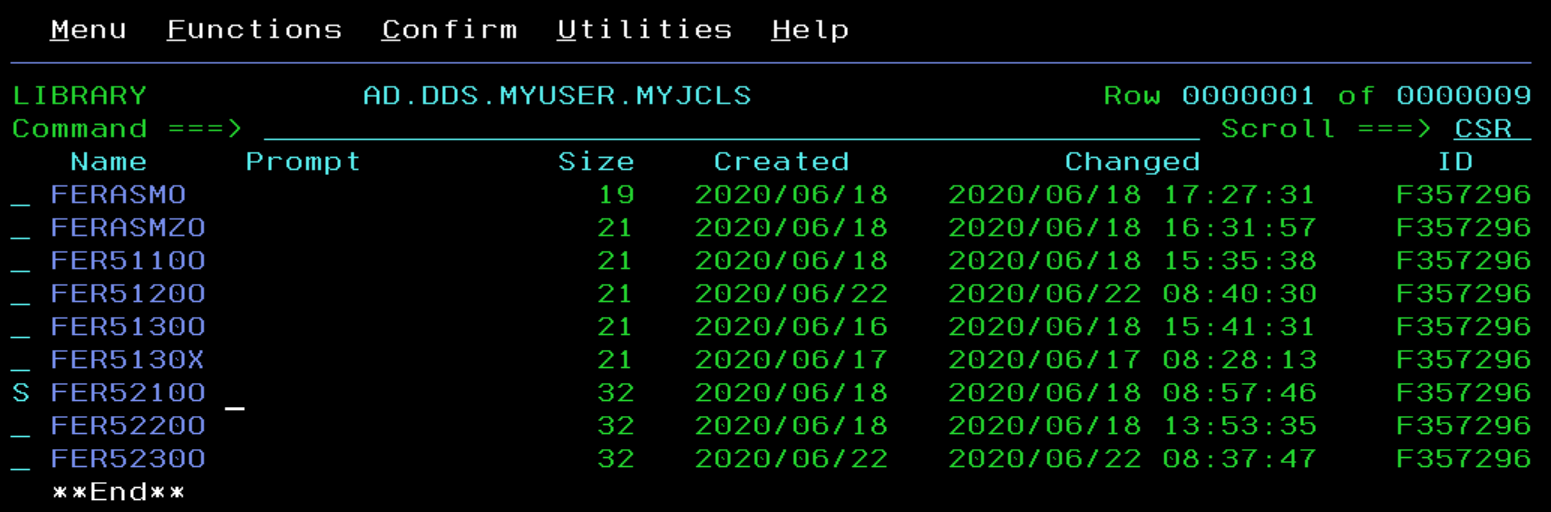
Na tela abaixo, digite “COPY” na linha de comando.



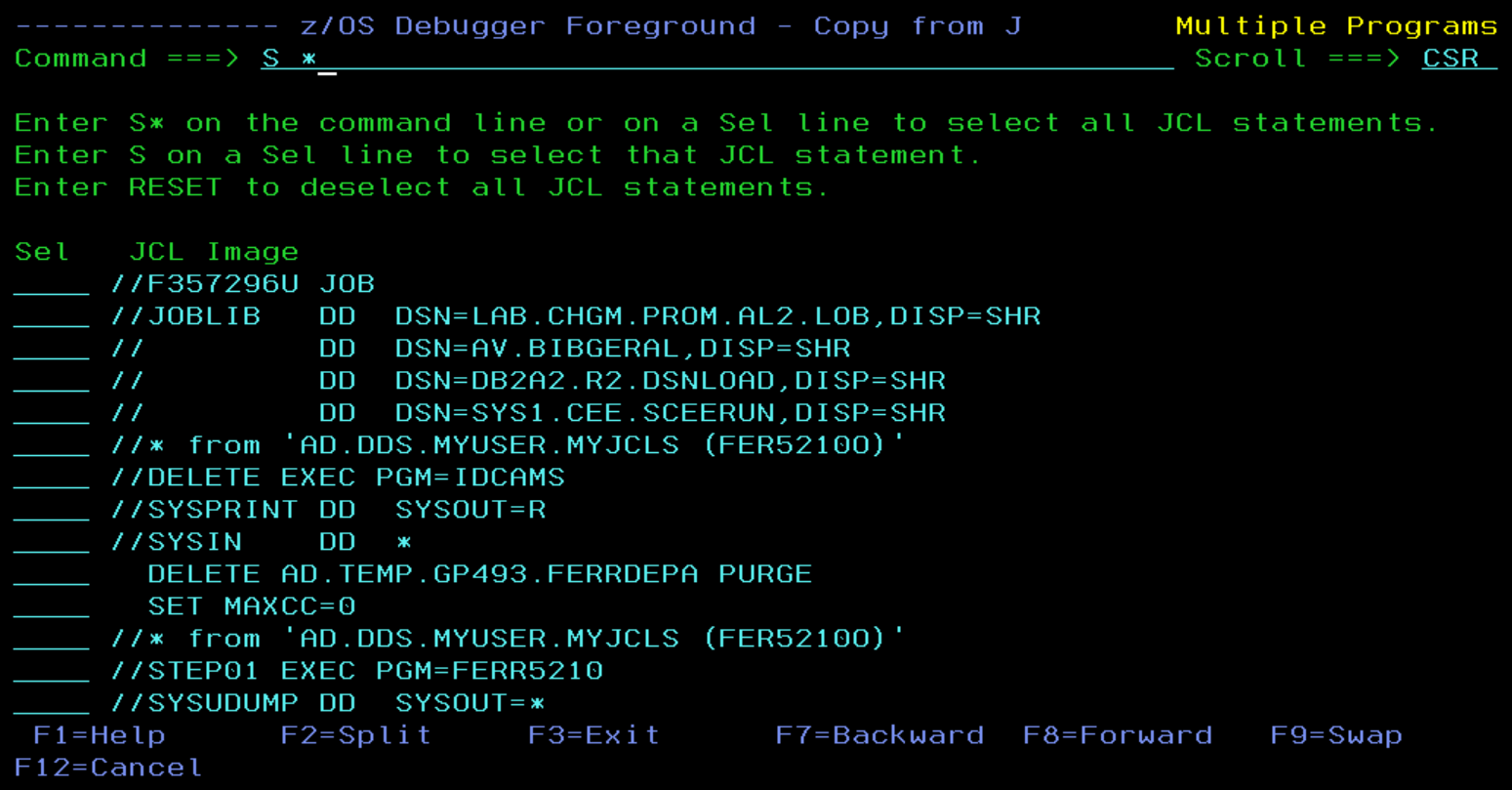
Informe o nome do arquivo onde encontra-se o JCL para a execução do programa, neste caso: AD.DDS.MYUSER.MYJCLS, seguido de <Enter>.

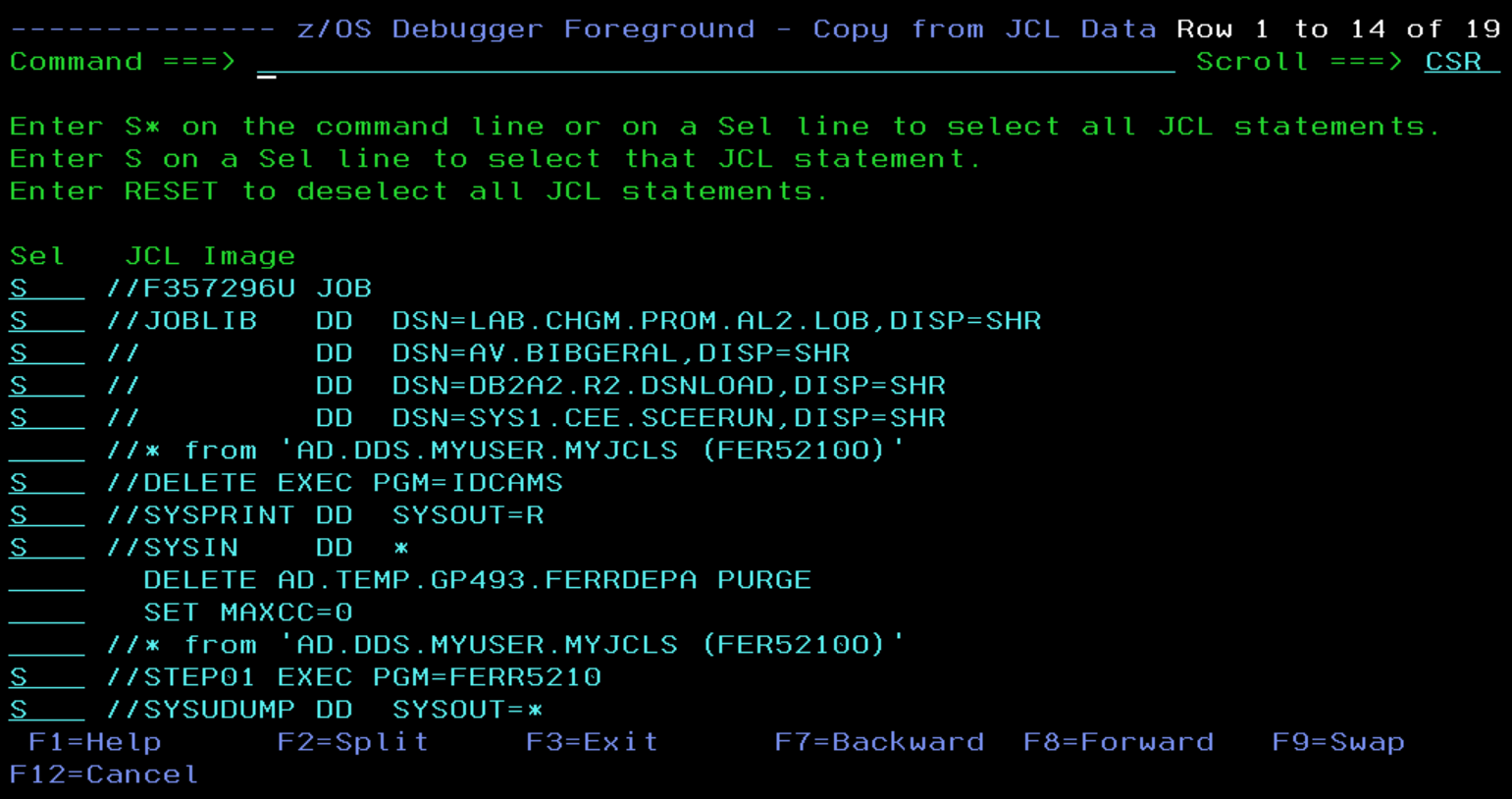


Selecione o membro com o JCL desejado, neste caso FERR5210O e <Enter>.

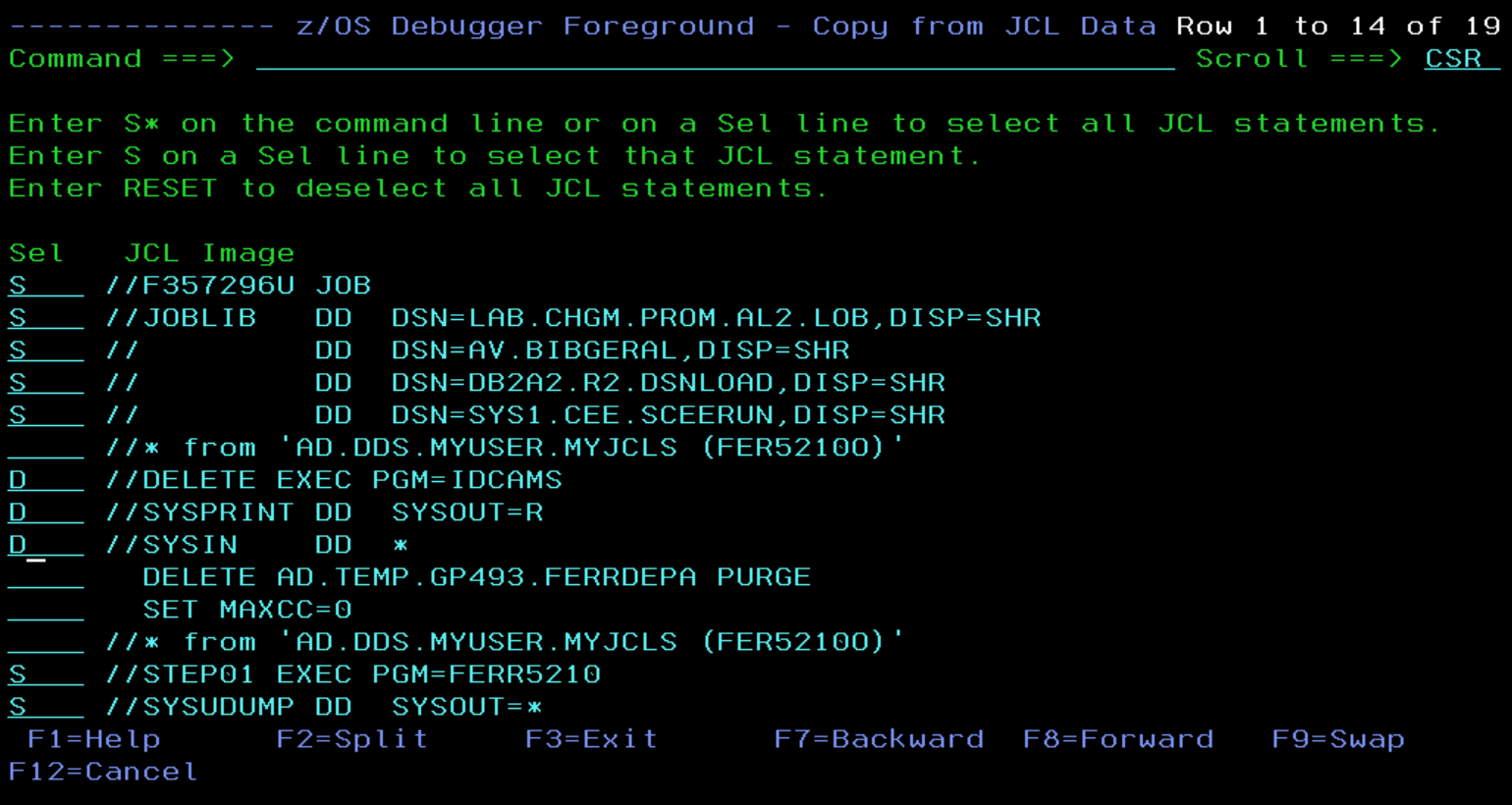


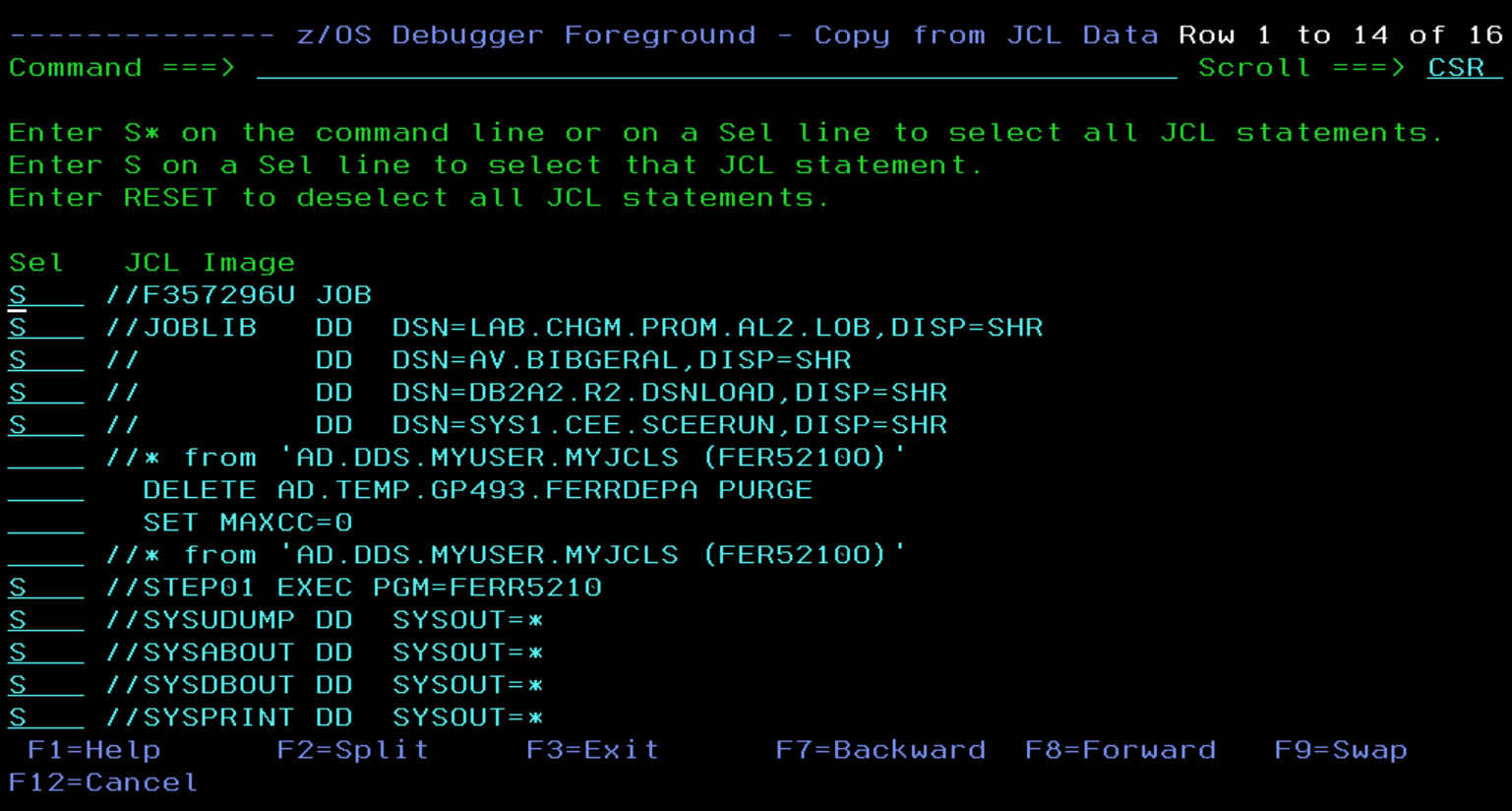
Será exibida uma tela com o JCL para a seleção das linhas necessárias para a execução, utilize “S \*” para selecionar todas as linhas, e em seguida iremos utiliza a opção “D” (delete) na coluna “Sel” para excluir as linhas não necessárias, note a mensagem “Multiple Programs” no canto superior direito, indicando que este JCL tem mais de um comando “EXEC PGM”, neste caso iremos retirar o IDCAMS.



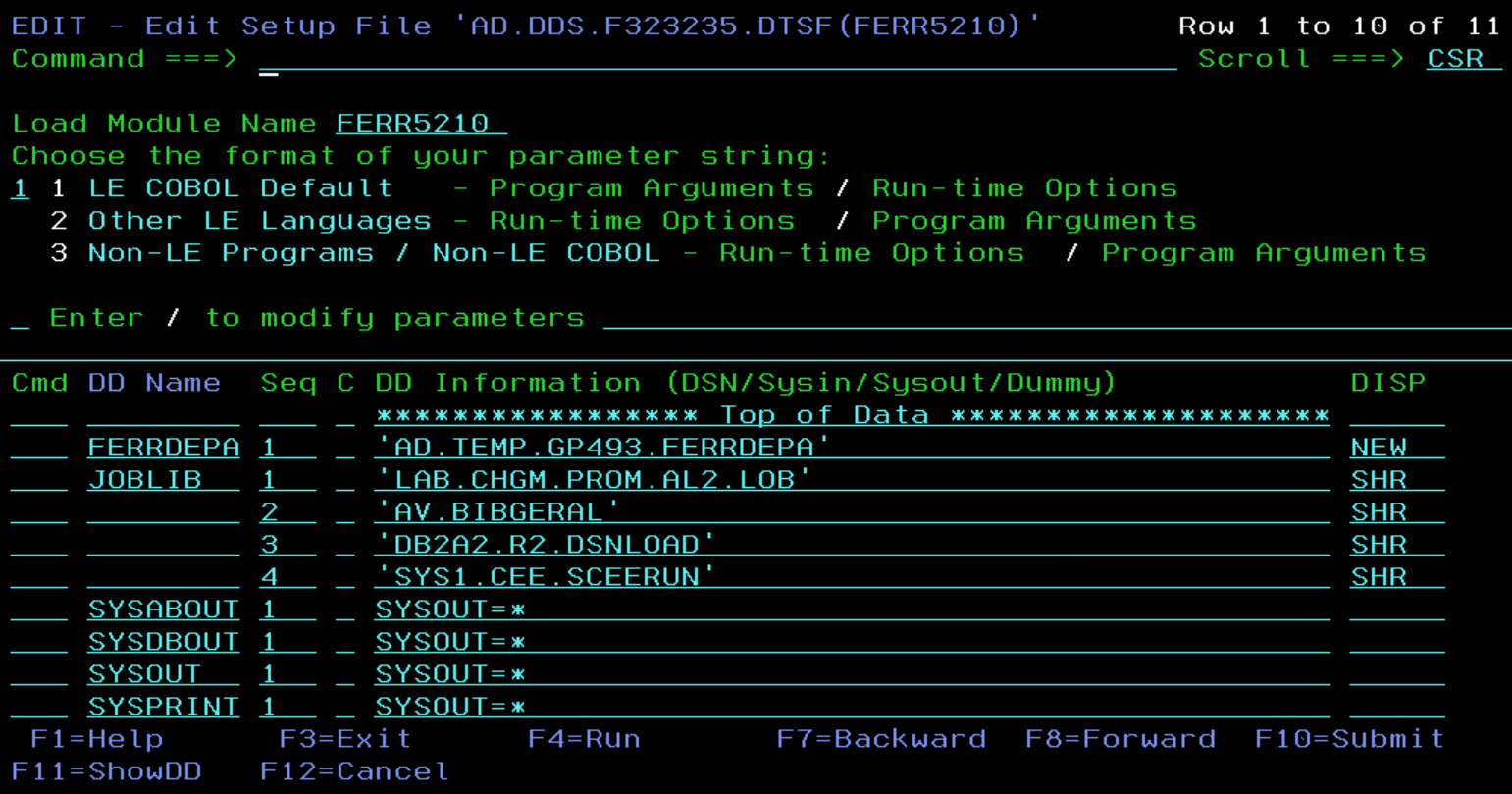


Faça a deleçaõ das linhas desnecessárias com “D” na coluna “Sel”, seguido de <enter>.





Digite <Pf3> para retornar a tla inicial.

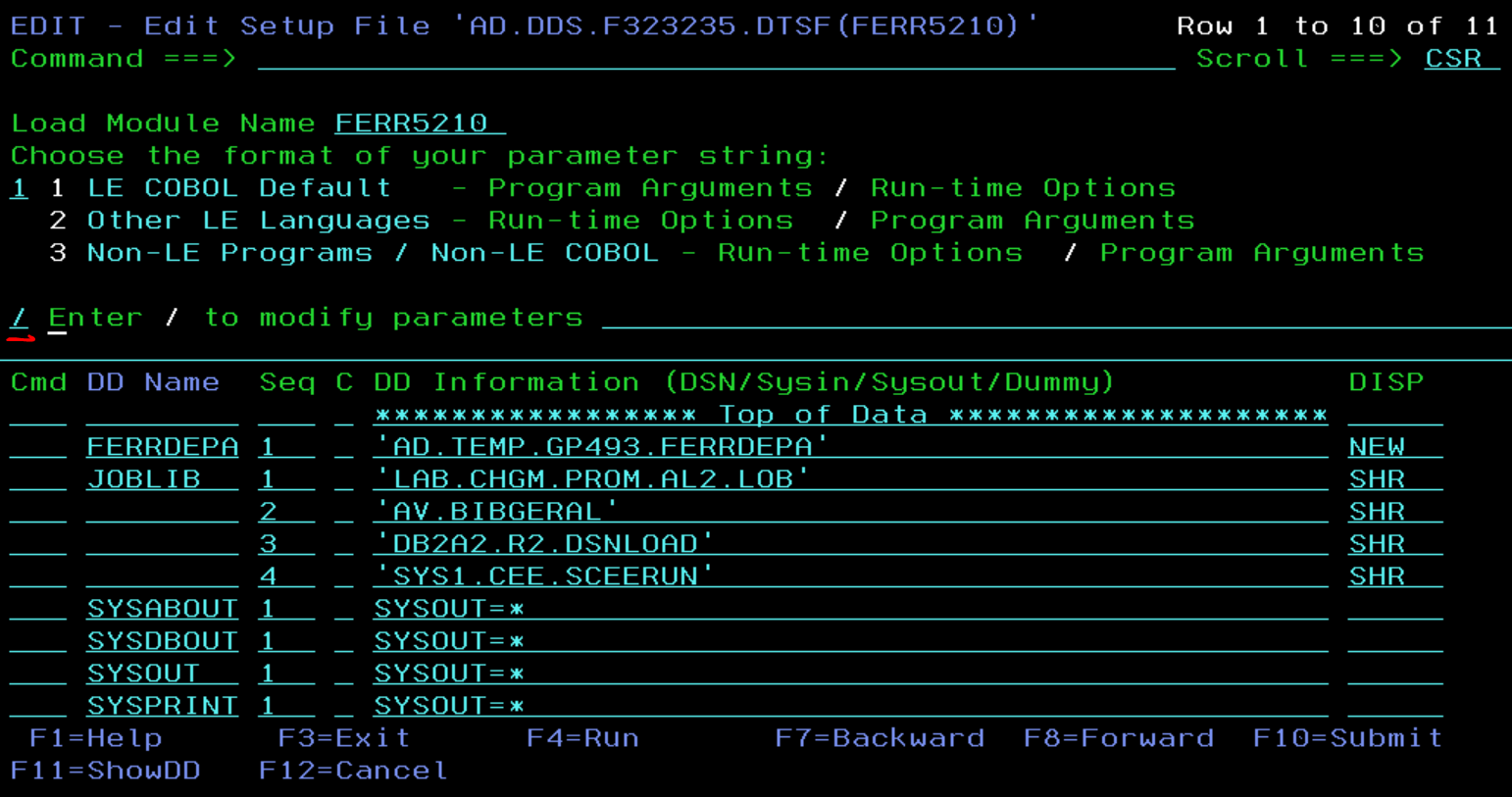


# **Informando parâmetros para a execução - Foreground.**

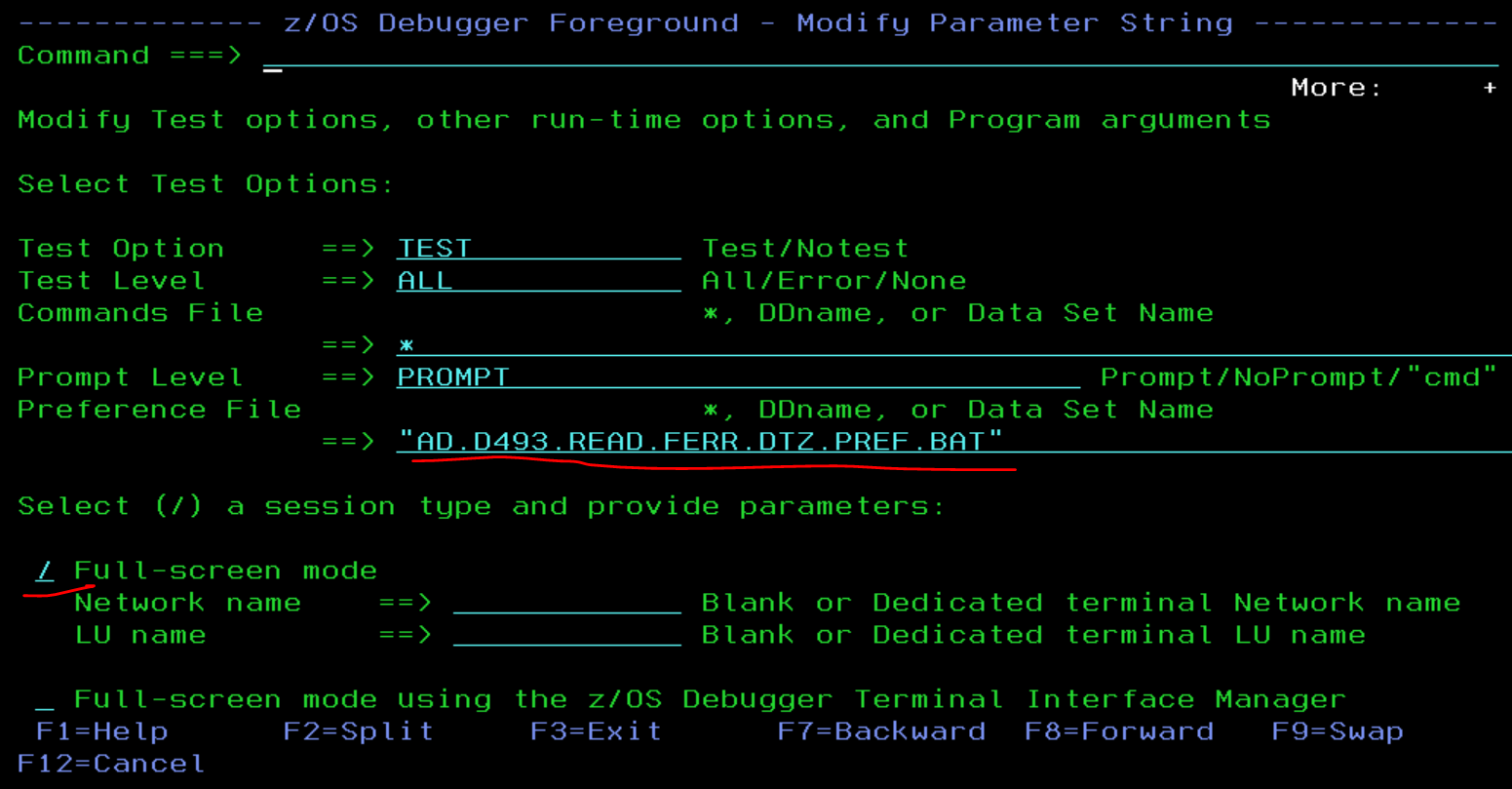
Necessitamos informar ao Debug Tool onde encontrar a listagem de compilação do seu programa, essa listagem é gerada no momento da compilação (stage) do Changeman quando marcamos com “/” o campo “Debug Tool”.

Não esqueça de fazer o Promote do programa, pois a versão do seu objeto e da listagem devem estar sincronizados.

Selecione com “/” o campo “Enter / to modify parameters”.



Informe o arquivo **"**AD.D493.READ.FERR.DTZ.PREF.BAT**"**, atenção sempre entre **“” (aspas duplas)** e selecione a opção “Full –screen mode” com “/”, digite <enter> e em seguida <Pf3> para sair.

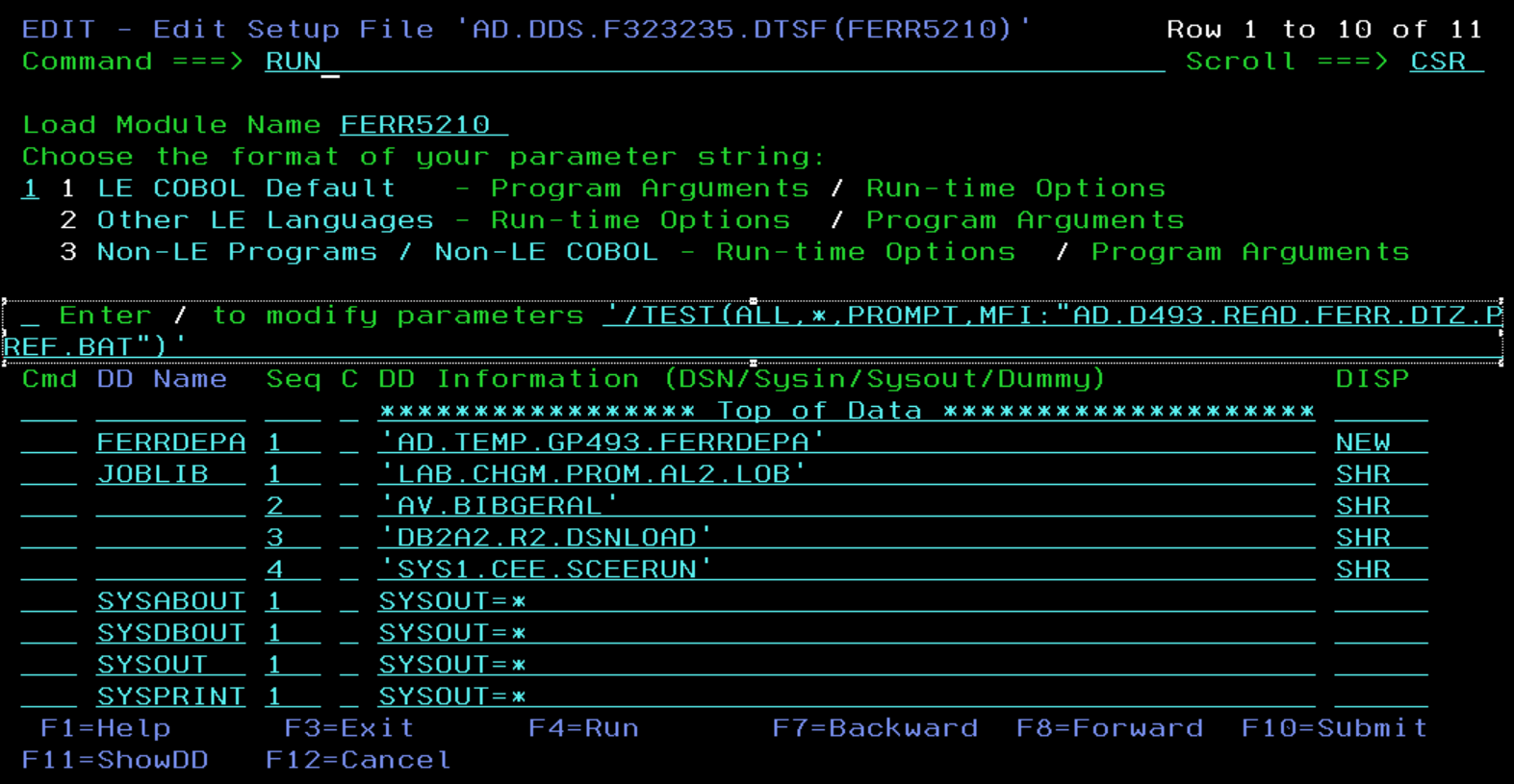


Note que o campo “modify parameters” foi alterado de acordo com a seleção feita.

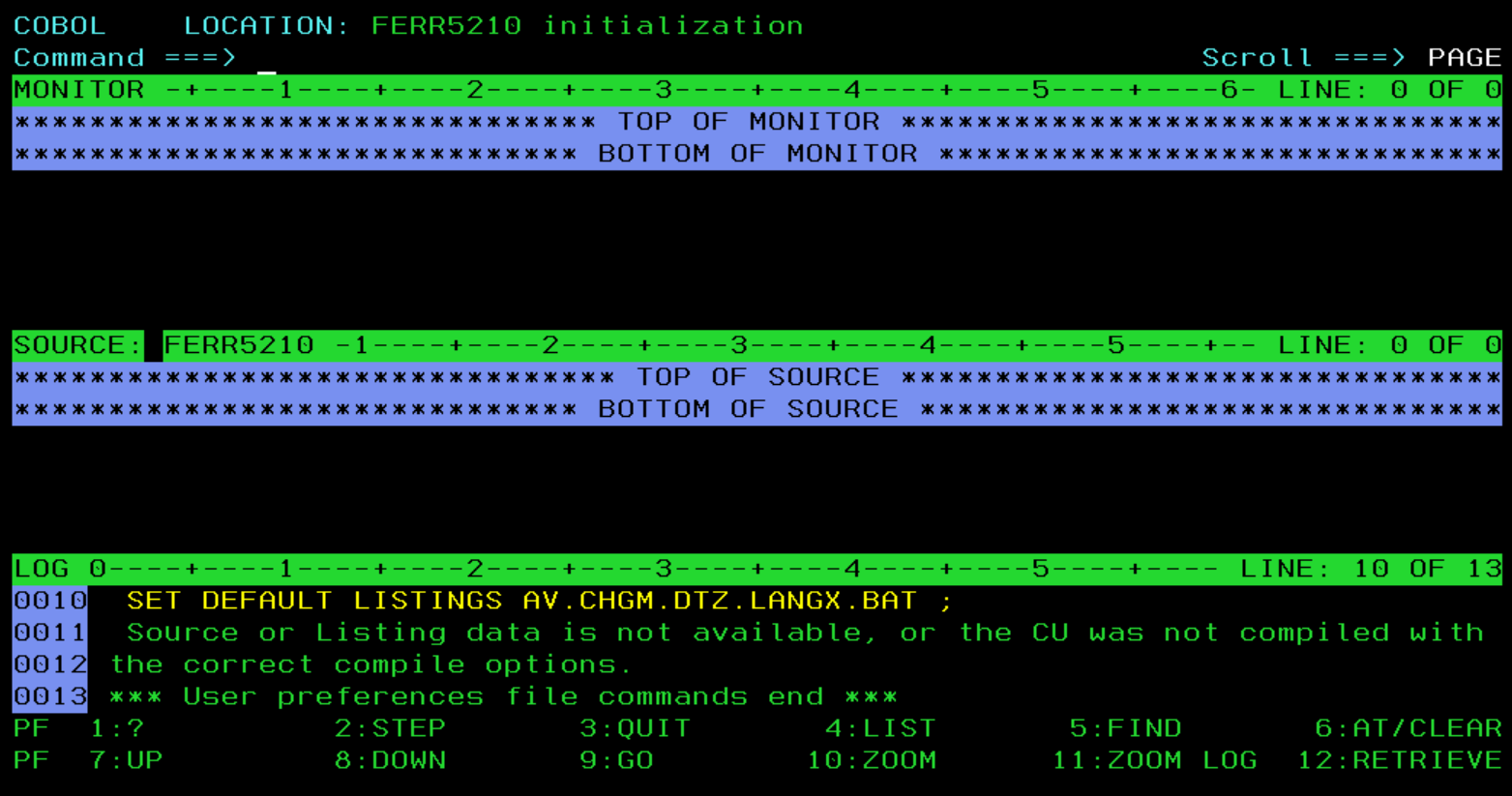


# **Executando um programa com o Debug Tool - Foreground.**

Digite “RUN” na linha de comando.



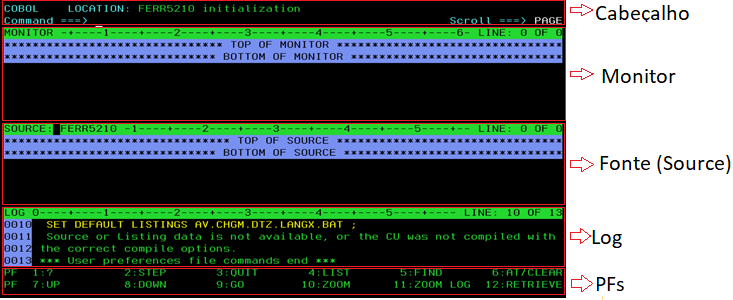
Será exibida a tela seguinte:



Para prosseguir com o Debug, veja o item “[Descrição das PF´s](#_Descrição_das_PF´s.)”.

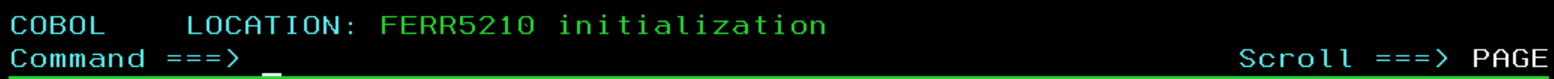
# **Painel padrão do Debug Tool.**

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.



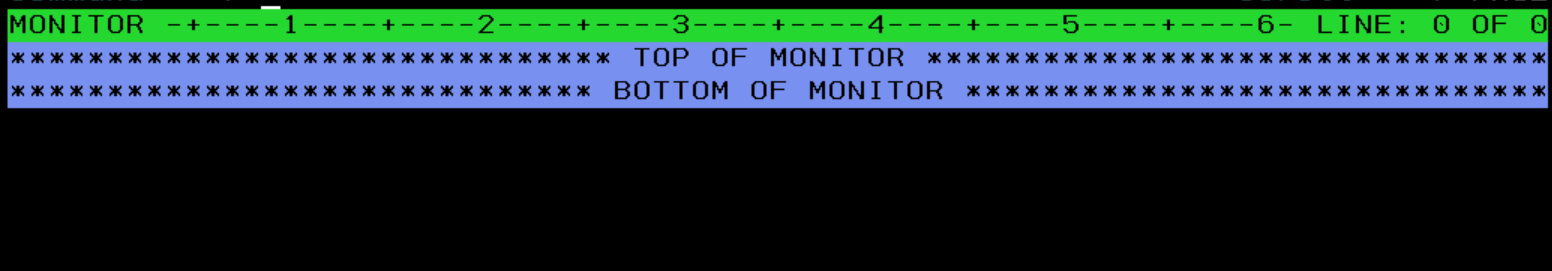
**Área de cabeçalho:**

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.



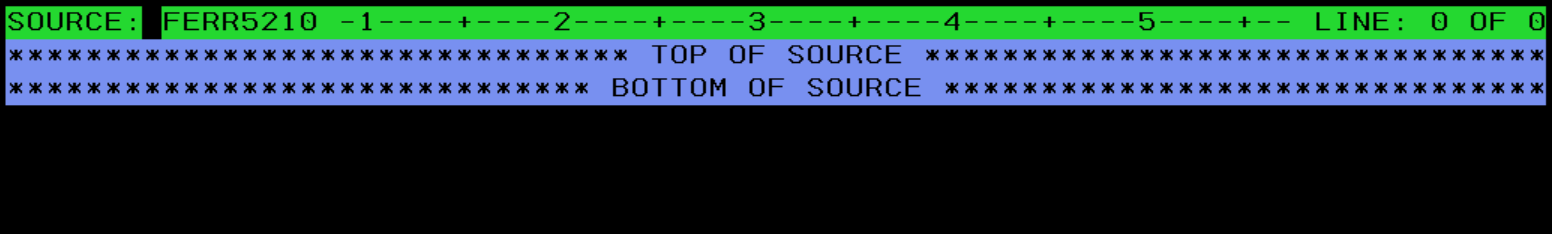
**Janela Monitor:**

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”.



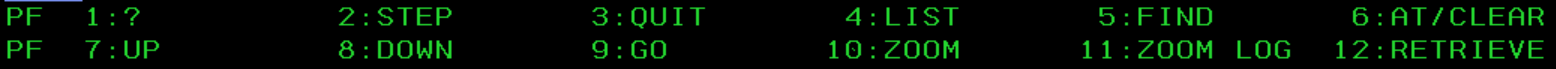
**Janela Log:**

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.

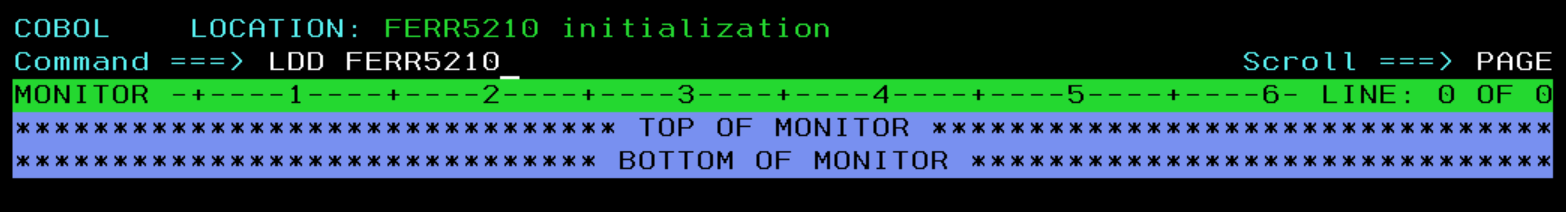


Mais adiante descreveremos as funções da PF´s e alguns outros comandos.

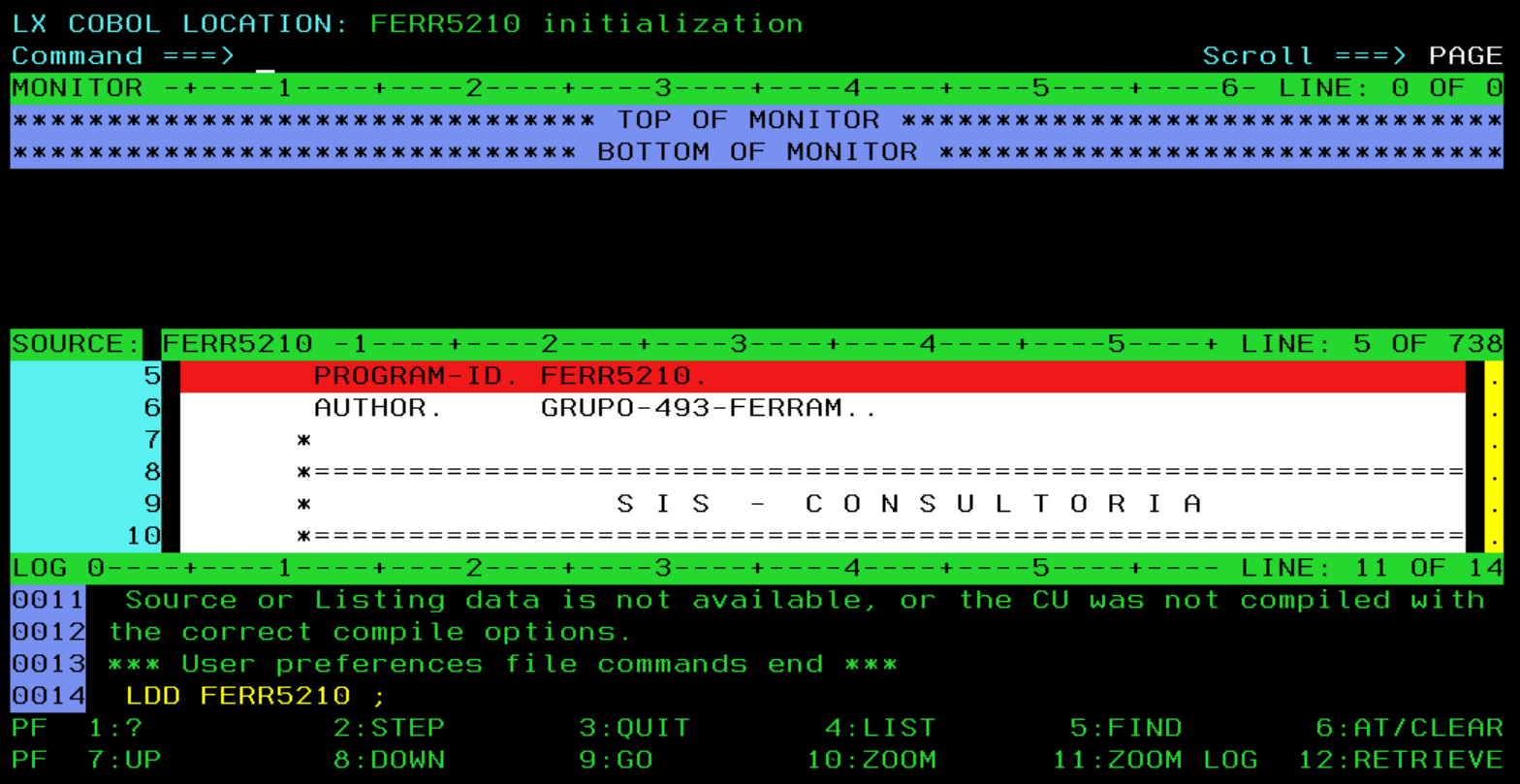
# **Carregando a listagem de um programa.**

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.

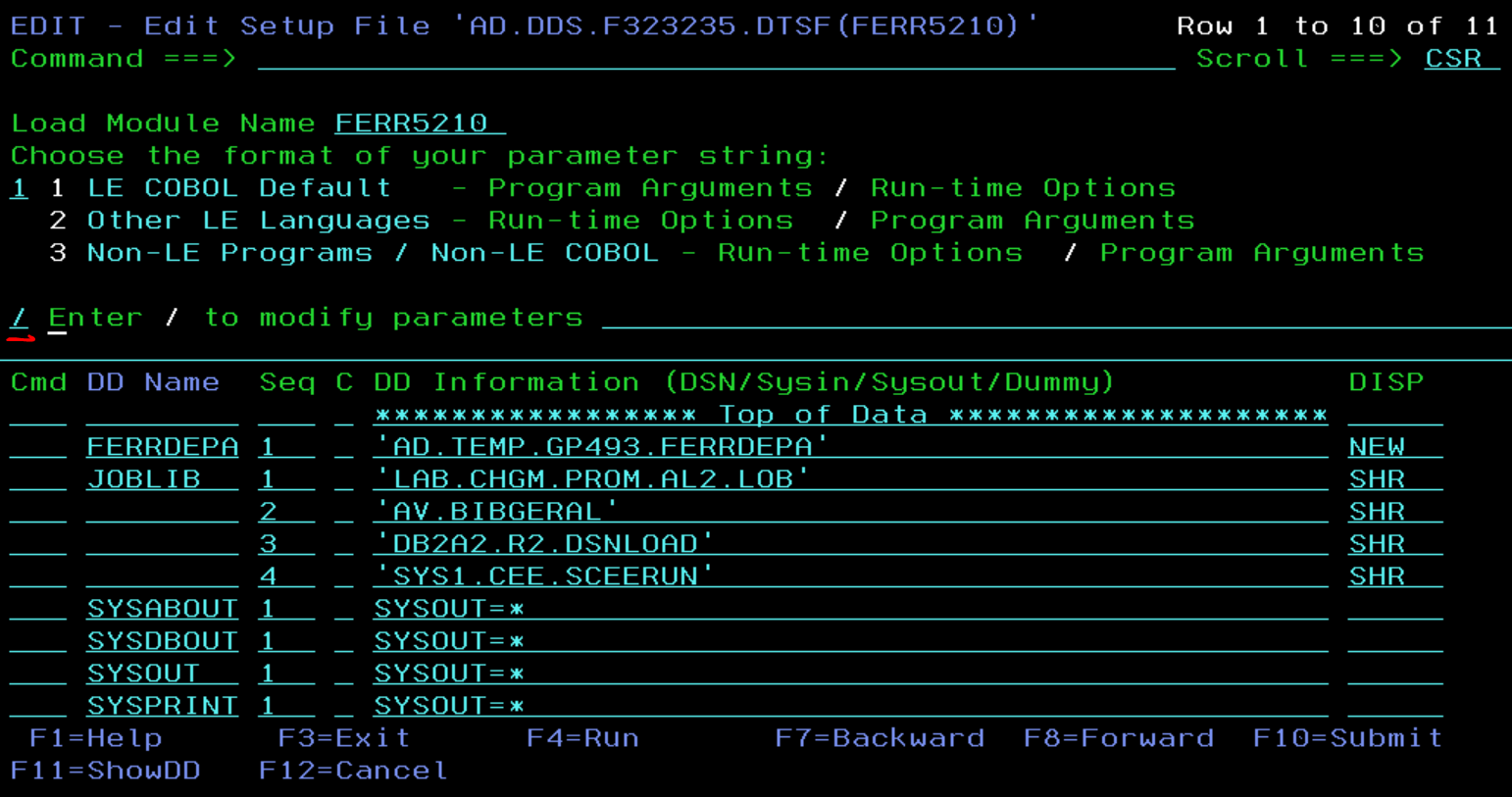


# **Informando parâmetros para a execução - Background.**

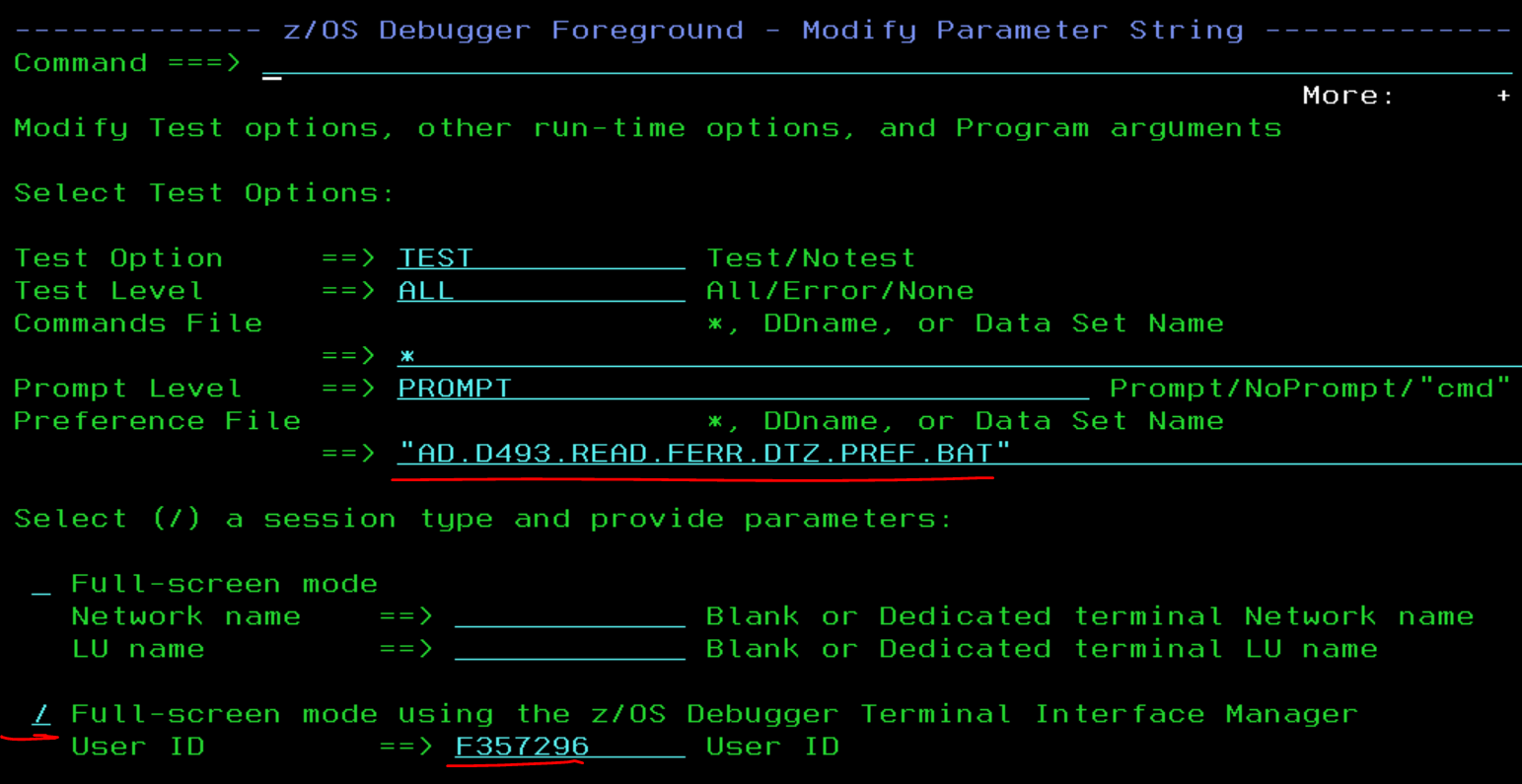
Necessitamos informar ao Debug Tool onde encontrar a listagem de compilação do seu programa, essa listagem é gerada no momento da compilação (stage) do Changeman quando marcamos com “/” o campo “Debug Tool”.

Não esqueça de fazer o Promote do programa, pois a versão do seu objeto e da listagem devem estar sincronizados. (conforme descrito no item [Preparando programa para Debug no Changeman](#_Preparando_programa_para).)

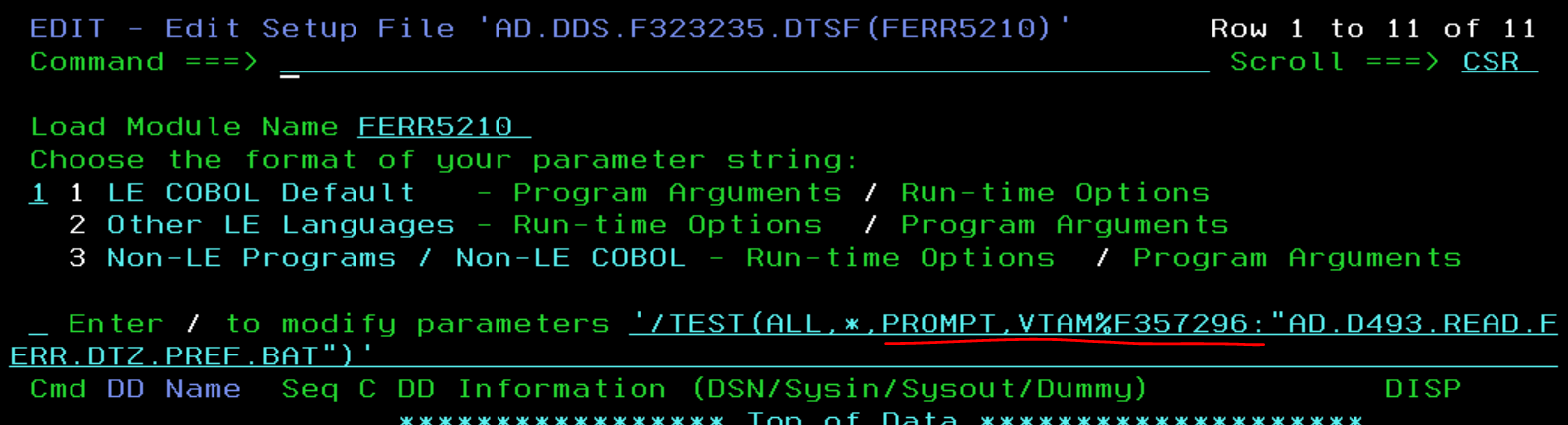
Selecione com “/” o campo “Enter / to modify parameters”.



Informe o arquivo **"**AD.D493.READ.FERR.DTZ.PREF.BAT**"**, atenção sempre entre **“” (aspas duplas)** e selecione a opção “Full –screen mode using z/OS Debugger Terminal Interface” com “/”, informe a chave do usuário no campo “User ID”, digite <enter> e em seguida <Pf3> para sair.



Note que o campo “modify parameters” foi alterado de acordo com a seleção feita.



Neste caso o JOB de execução será submetido por essa sessão TSO e o Debug será feito através do TIM (Terminal Inteface Manager) associado ao “User ID” informado.

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

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.

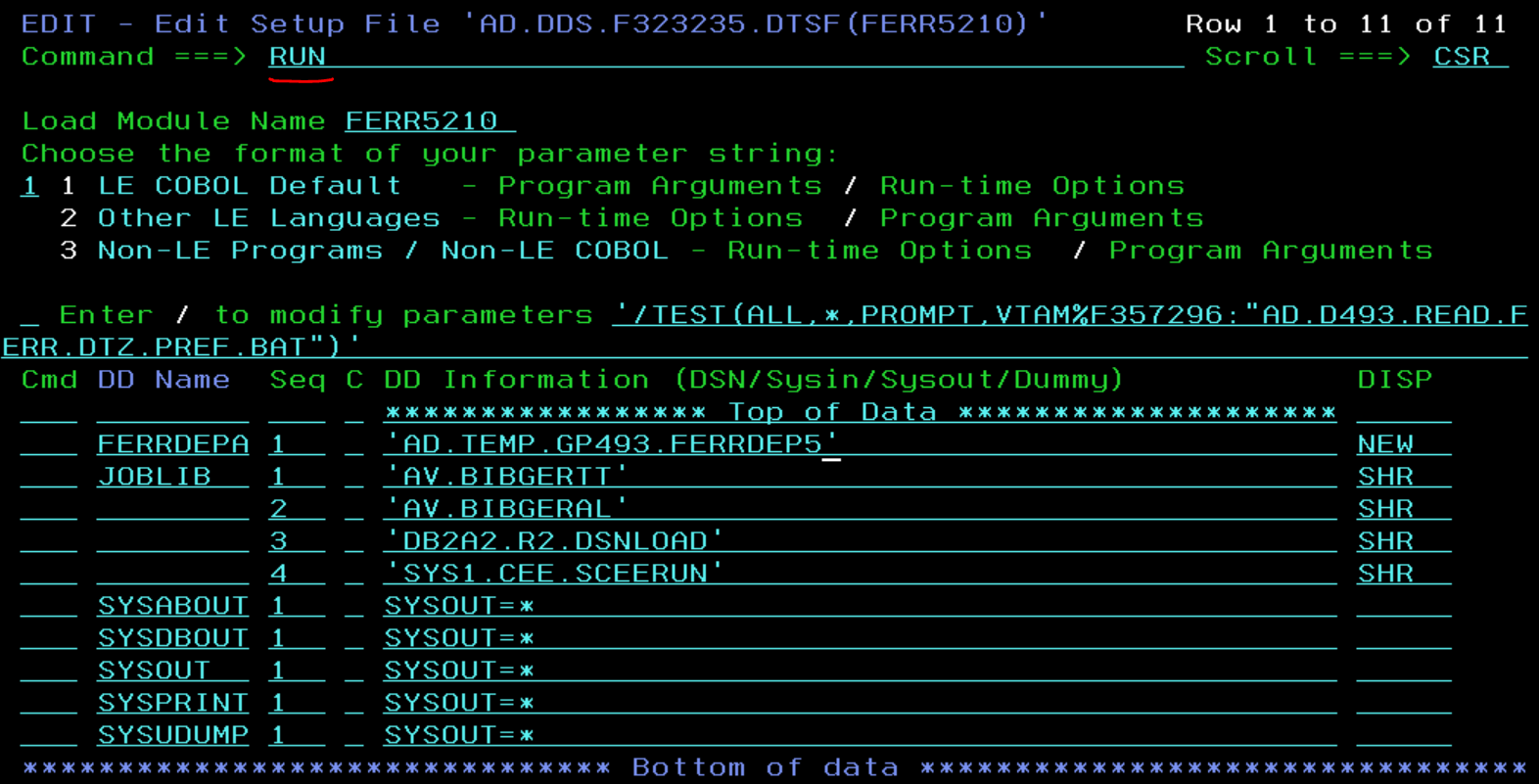


Deixar nesta tela e retornar a outra sessão de terminal:



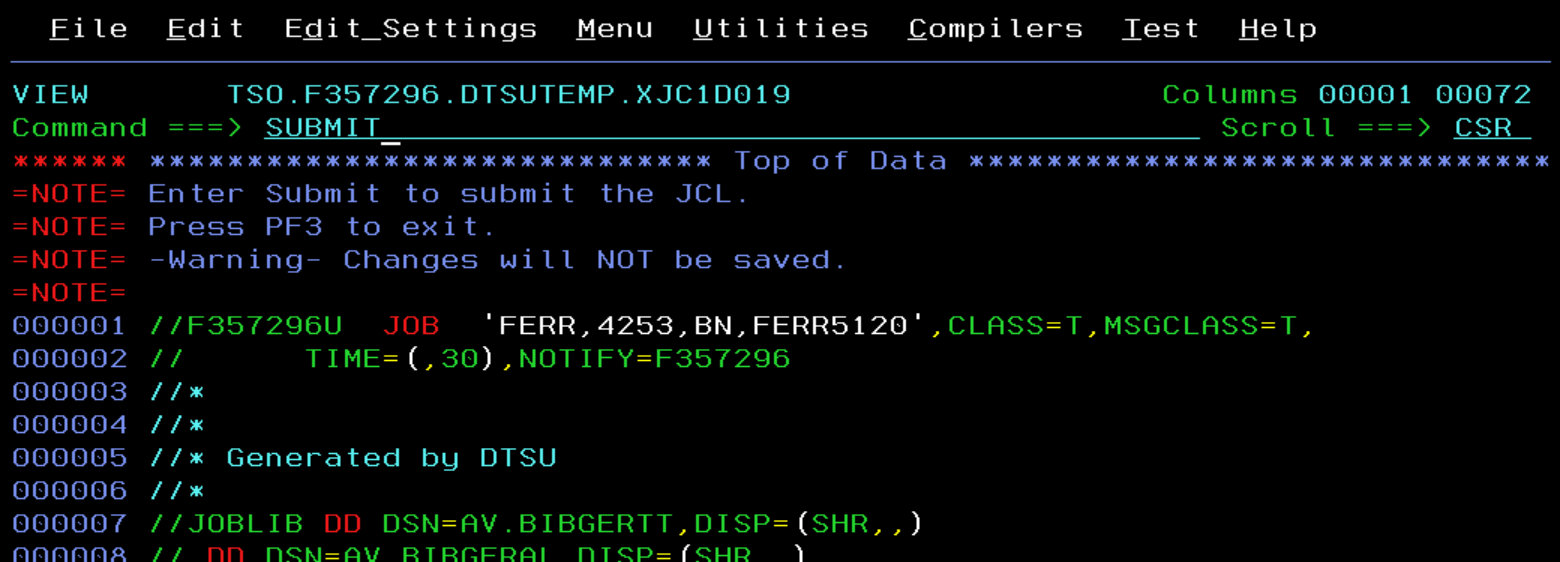
# **Executando um programa com o Debug Tool - Background.**

Na sessão TSO fazer a execução do programa com o comando RUN (ou <Pf4>) que mantém a sessão TSO presa enquanto durar o Debug, ou através do comando SUBMIT (OU <Pf10>) que libera o TSO.



Voltar para a sessão onde o TIM está aberto, deverá estar na tela do Debug.

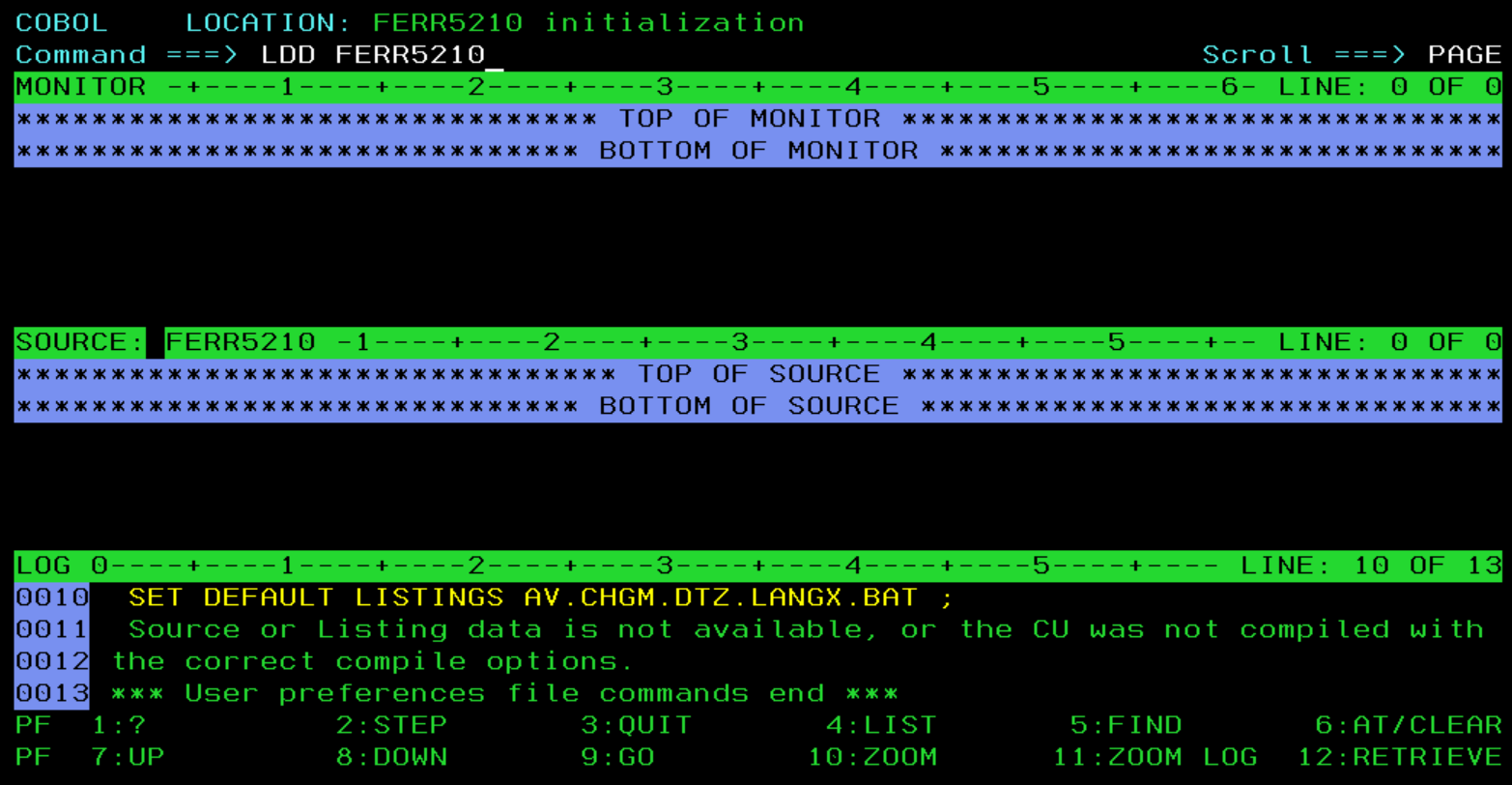
Caso tenha feita a opção pelo comando SUBMIT (OU <Pf10>), será aberta a tela abaixo com o JCL para execução, que deve ser comandado através do SUBMIT.



Para acompanhar a execução volte para a sessão onde está aberto o TIM.

Programas compilados com 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).

Já se estiver compilado com Enterprise Cobol v5 (ou superior) a listagem será exibida diretamente sem a necessidade do LDD.



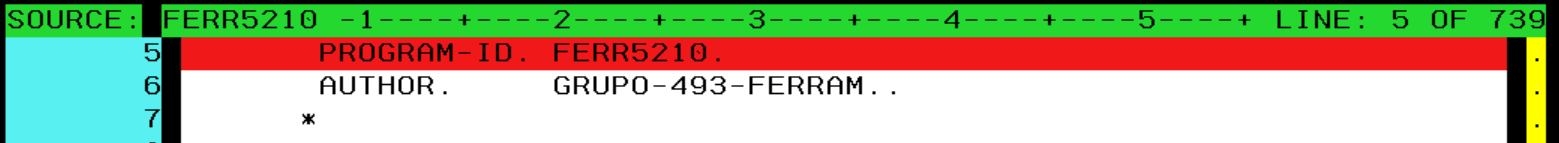
Dar continuidade no Debug utilizando os comandos e PFs descritos no item [Descrição das PF´s](#_Descrição_das_PF´s.).

# **Descrição das PF´s.**

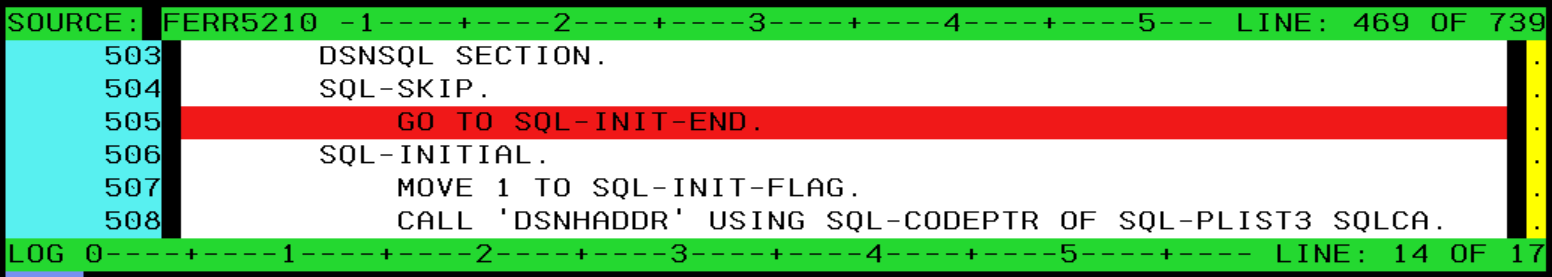
**PF2-STEP:**

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.

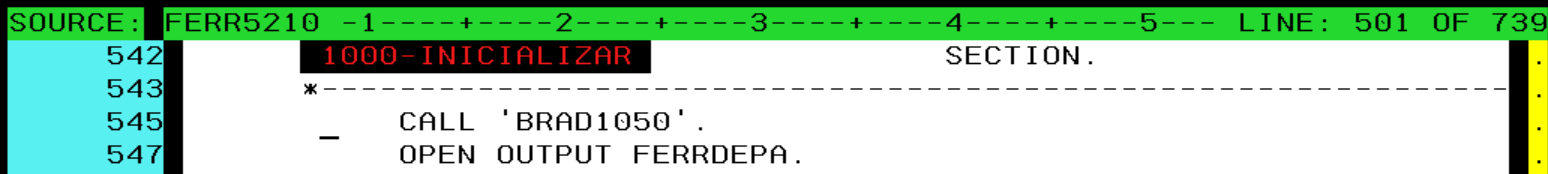


Após acionar <Pf2> 3 vezes o programa é posicionado na primeira instrução da PROCEDURE DIVISION.

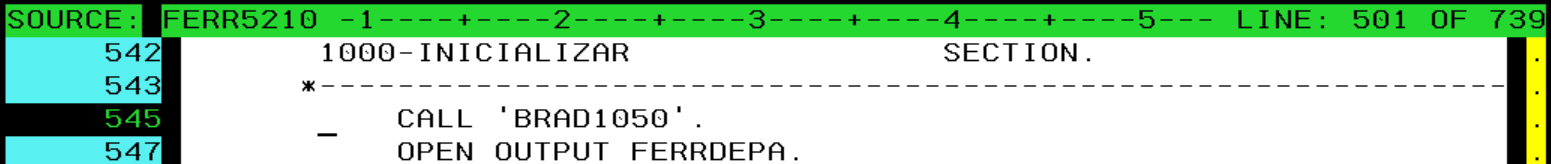


**PF6-AT/CLEAR:**

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

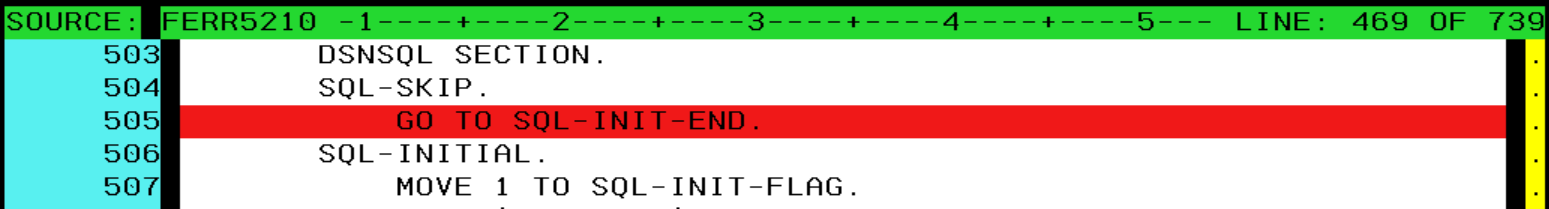


Após <Pf6>, note que a linha 545 onde o curso estava ficou em “destaque” indicando um BREAK POINT.



Para retornar ao inicio da PROCEDURE DIVISION, podemos utilizar várias <Pf8> ou o comando “QUALIFY RESET” (ou “QUA RES”).

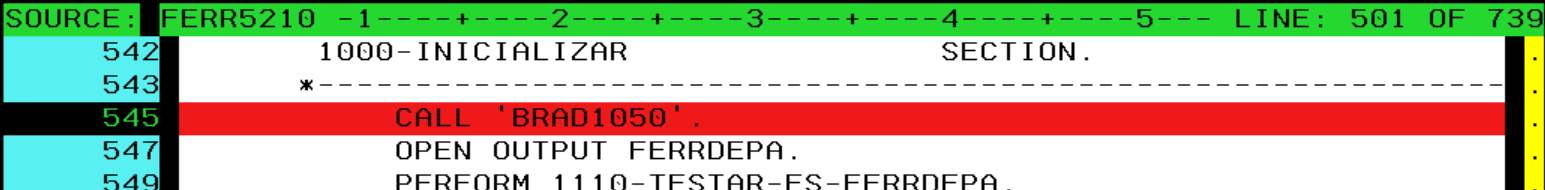




**PF9-GO:**

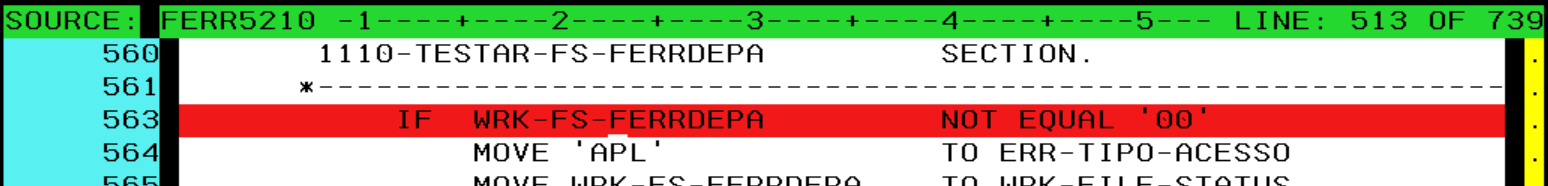
Executa o programa até o próximo BREAK POINT ou até o final caso não encontre BREAK POINT.

Após a <Pf9> o programa executou até o BREAK POINT da linha 545.

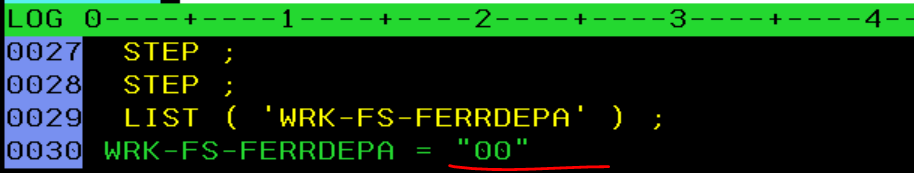


**PF4-LIST:**

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



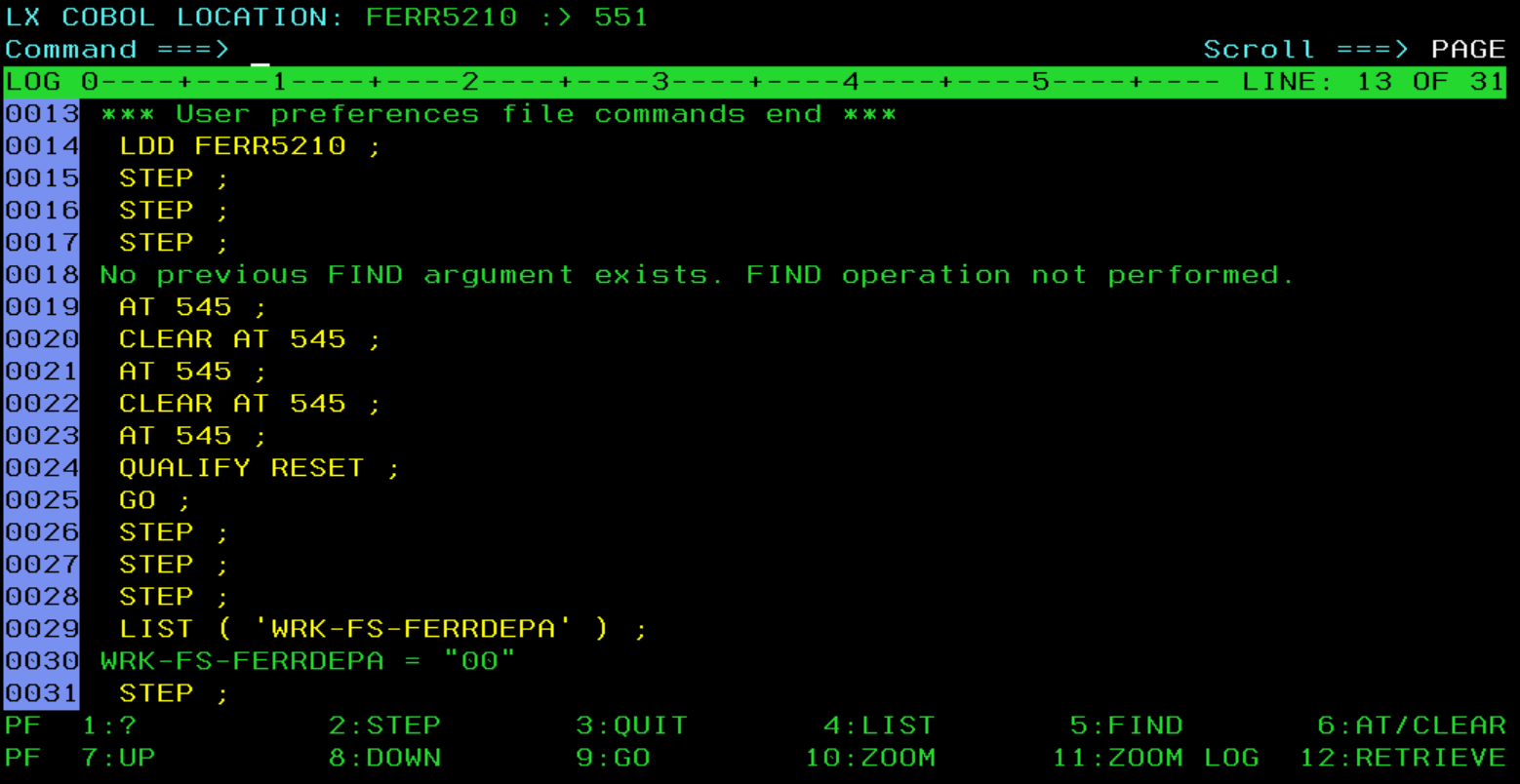
Após teclar o “<Pf4> - List” o conteúdo da variável é exibido na janela de Log.



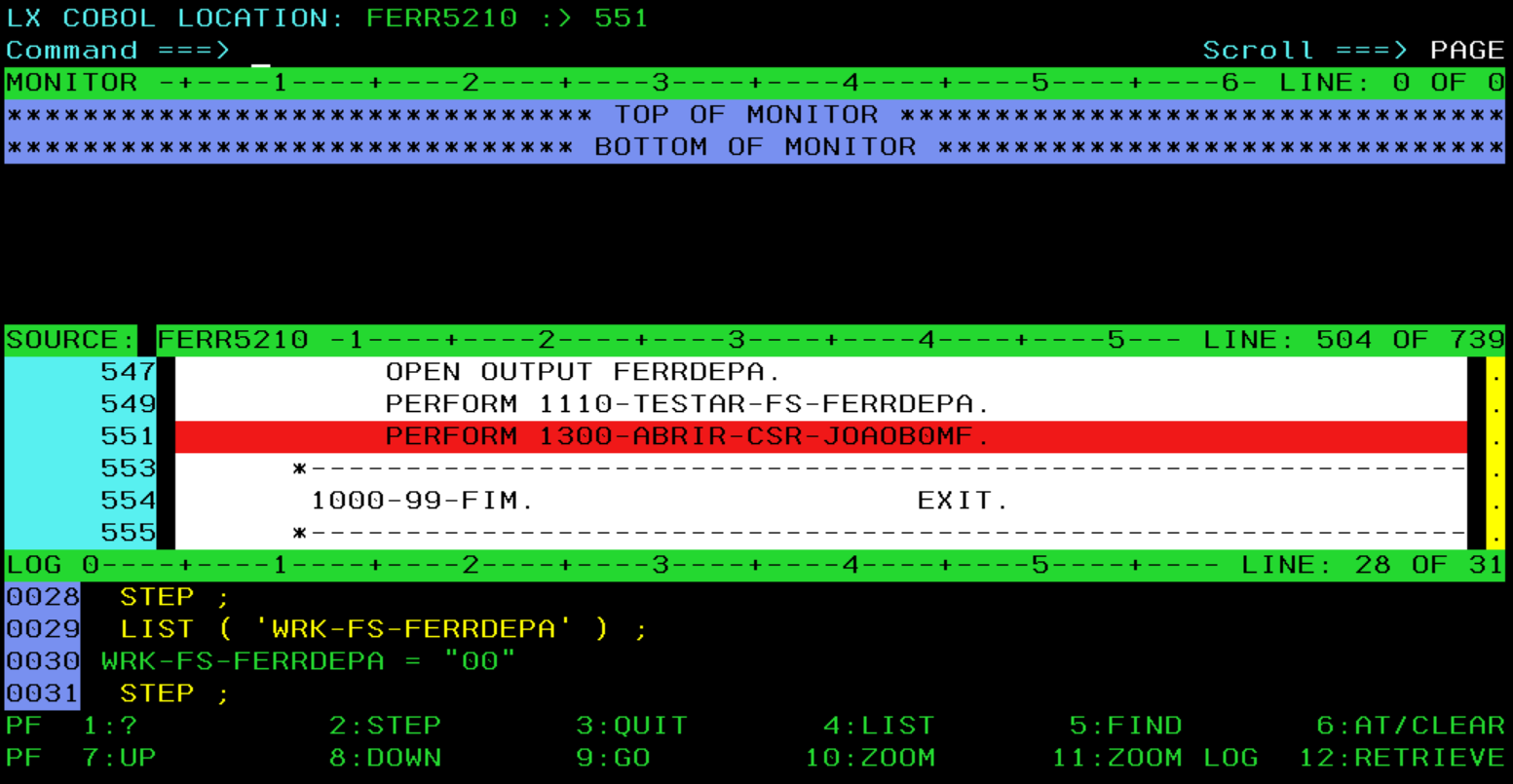
**PF11-ZOOM LOG:**

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.

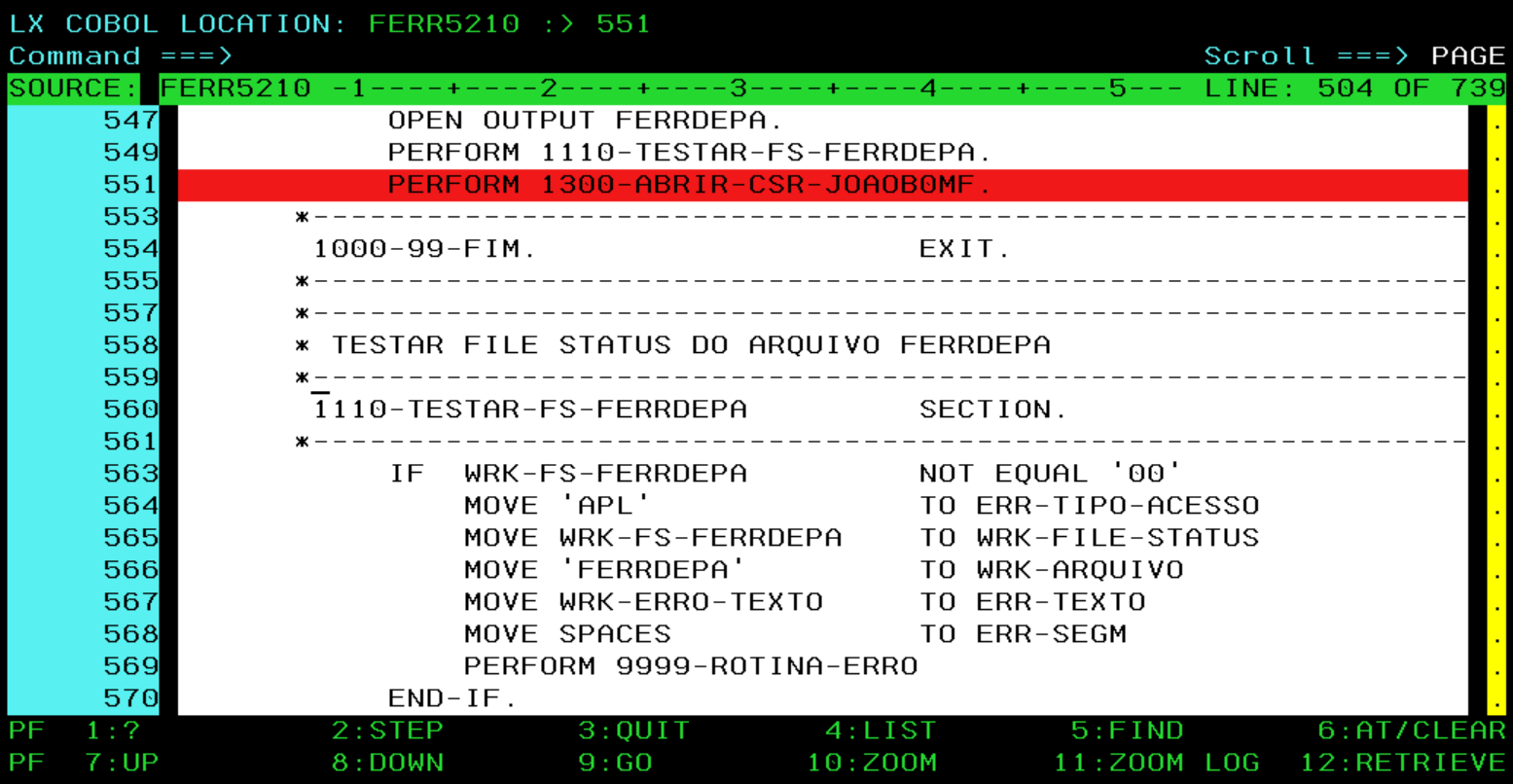


**PF10-ZOOM:**

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

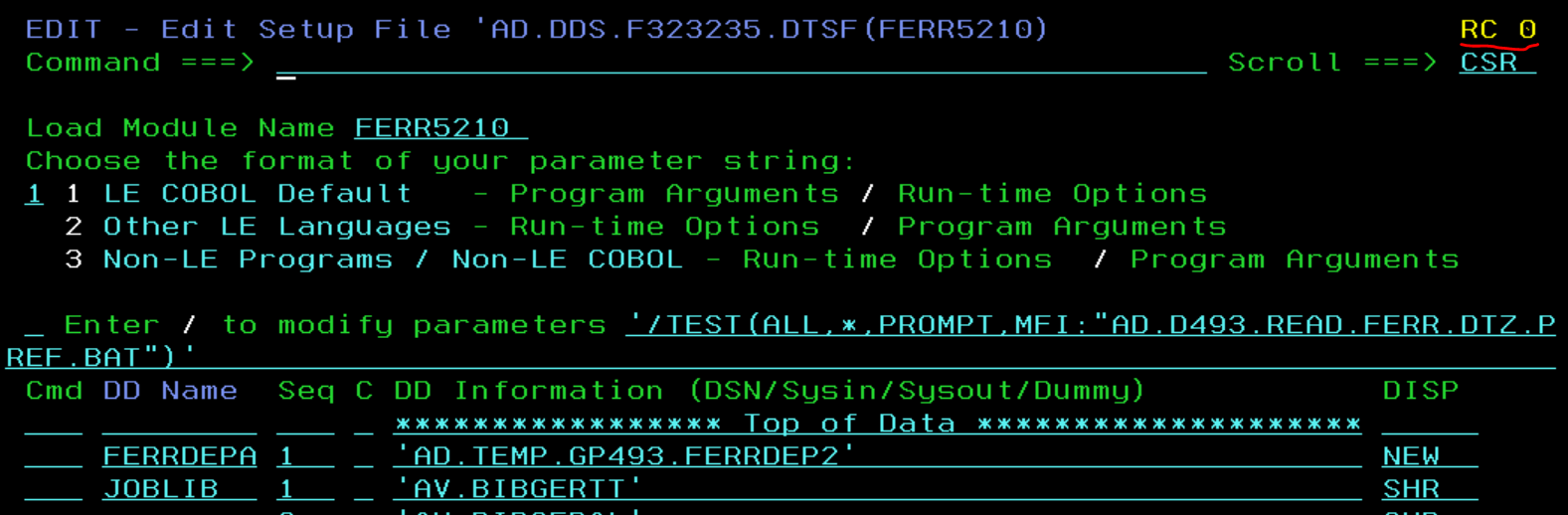


**PF3-QUIT:**

Encerra a sessão de Debug, termina o programa no ponto em que estava parado, não executa mais o programa.



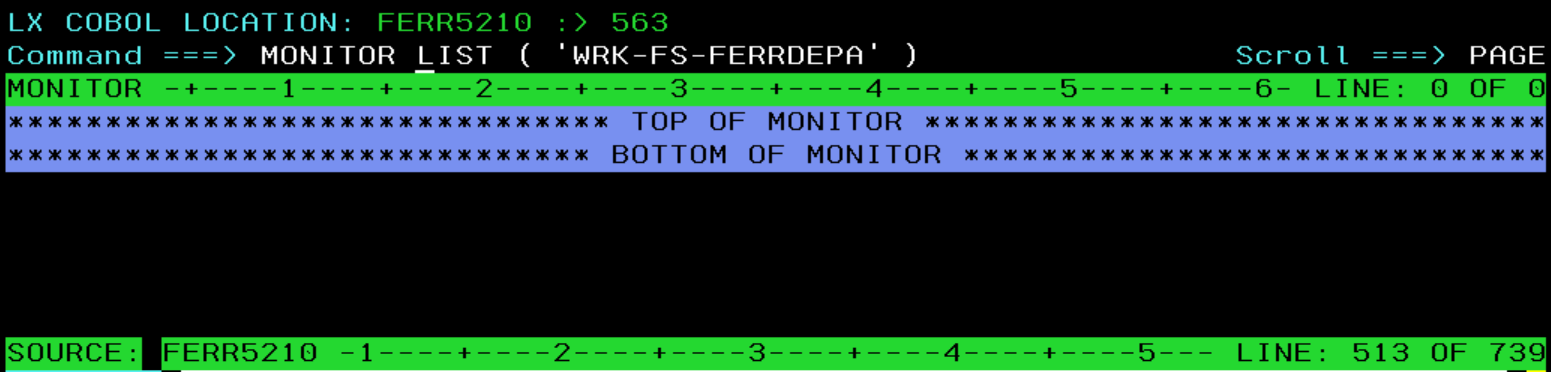
Retorna para a tela inicial.



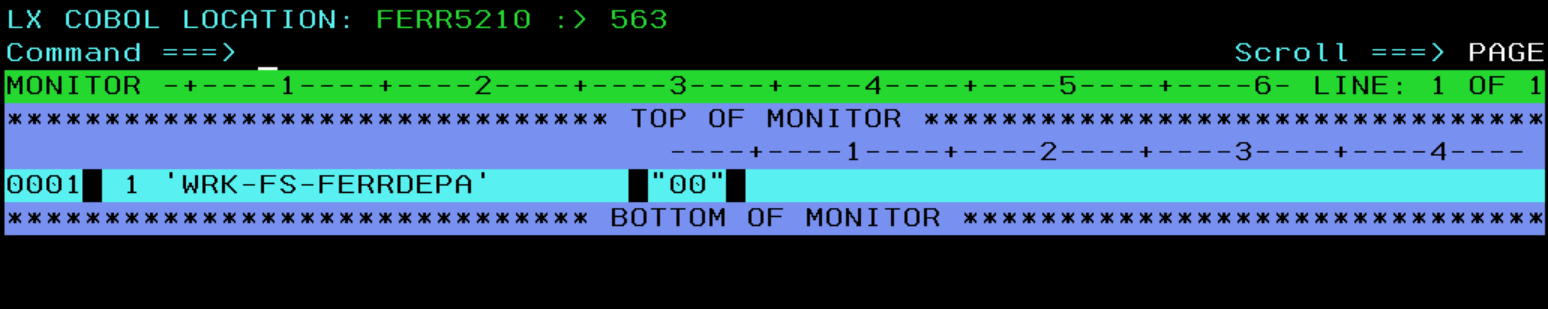
# **Monitorando variáveis.**

Modo I: Monitorando variáveis individualmente

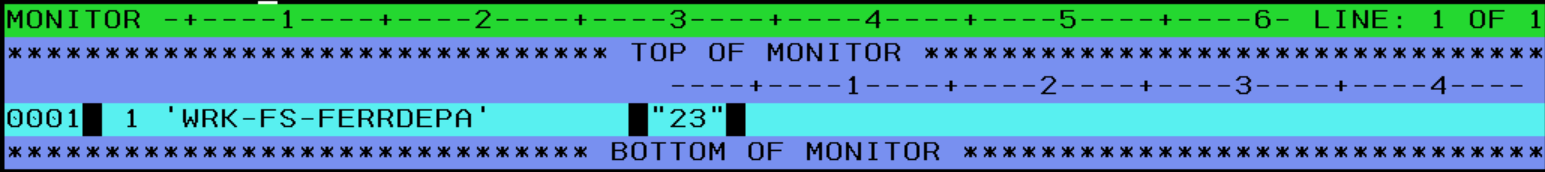
Entre com o comando MONITOR LIST ( 'nome-da-var' )



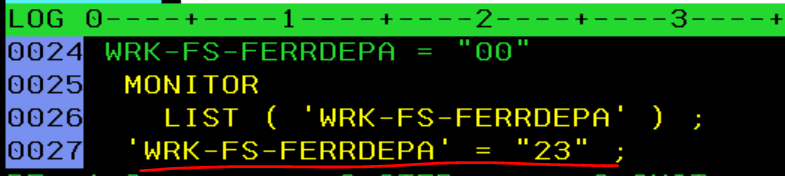
Será exibido o conteúdo da variável na janela Monitor, esta janela ficará fixa na tela sendo atualizada quando a variável sofrer atualização.



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.

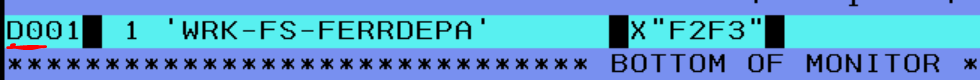


Exibindo o valor em Hexadecimal, digite “H” no número em frente ao nome da variável e <Enter>.





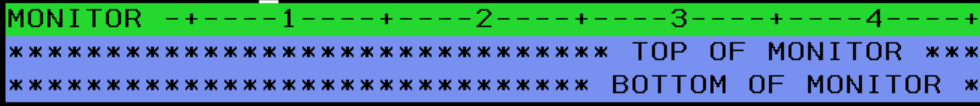
Para voltar ao modo “Default”, digite “D” no número em frente ao nome da variável e <Enter>.



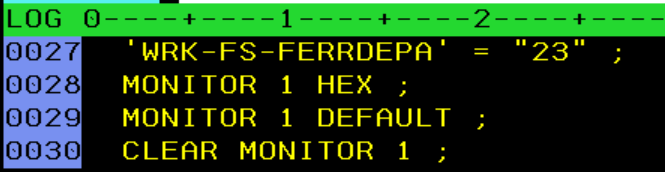


Digite “C” para fechar (“limpar”) o monitor de uma variável.





Note que na janela de Log aparecem os comandos correspondentes:



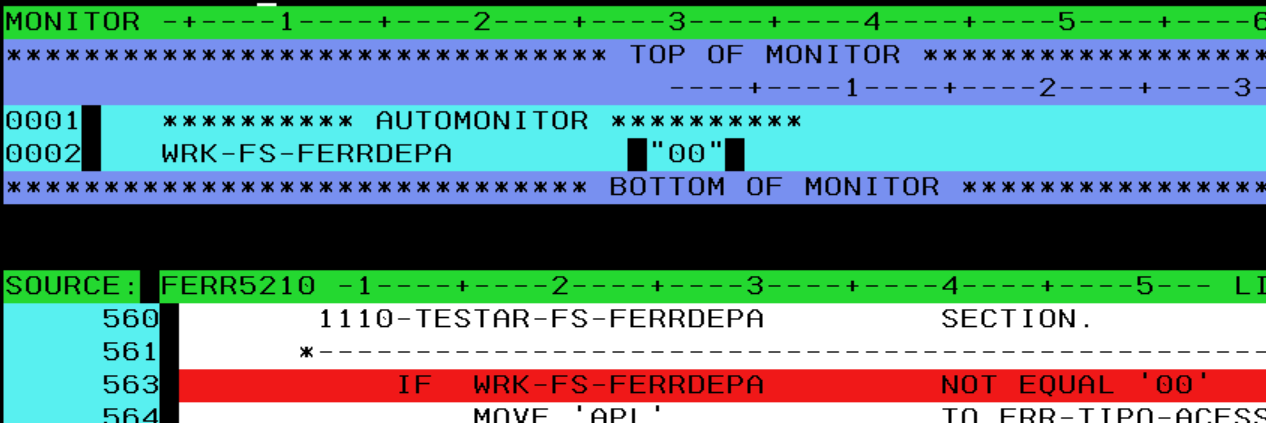
Modo II: Monitorando variáveis automaticamente

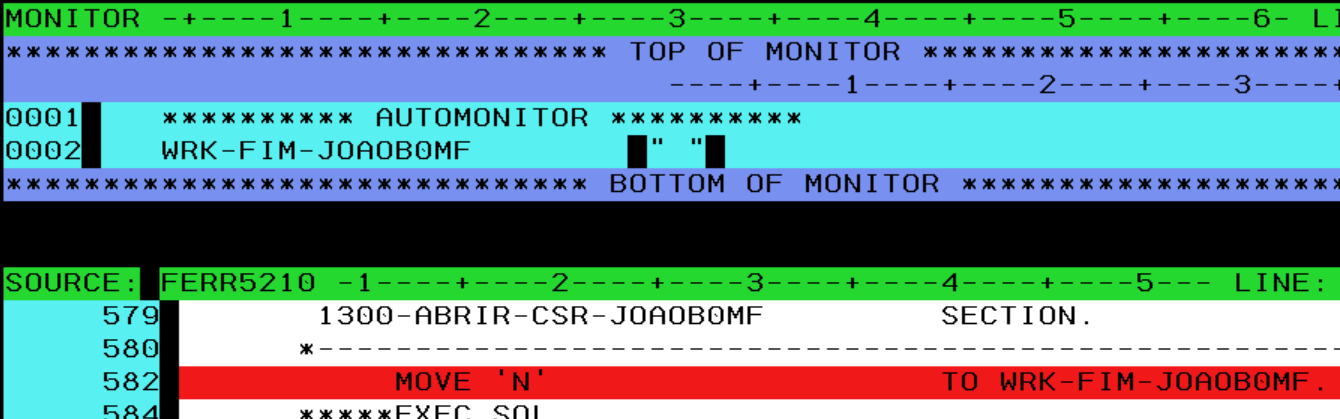
Neste caso, serão monitoradas automaticamente as variáveis envolvidas no comando executado.

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.

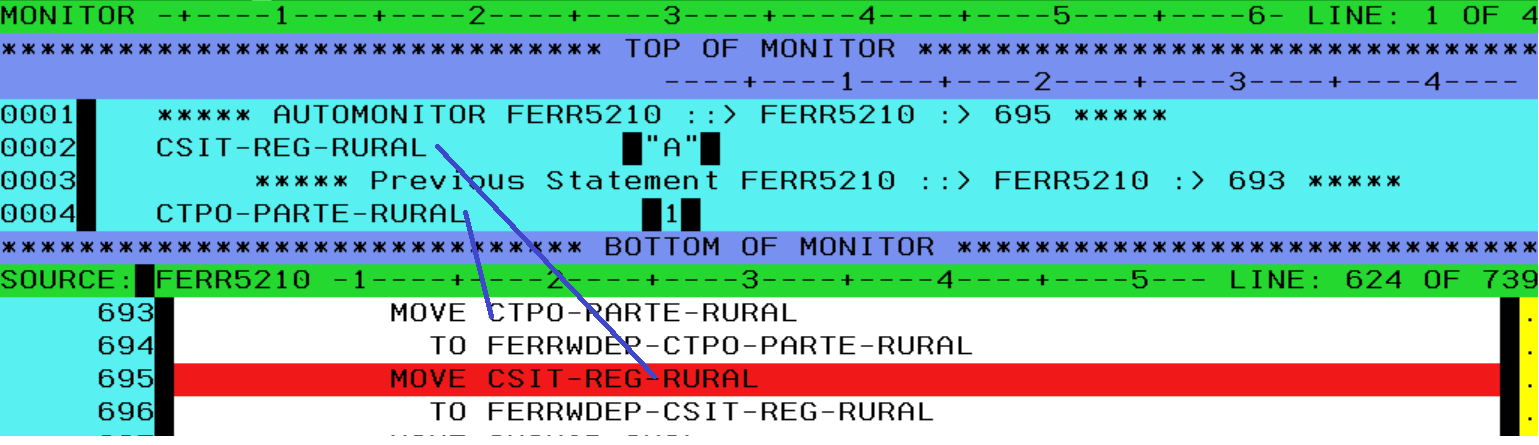




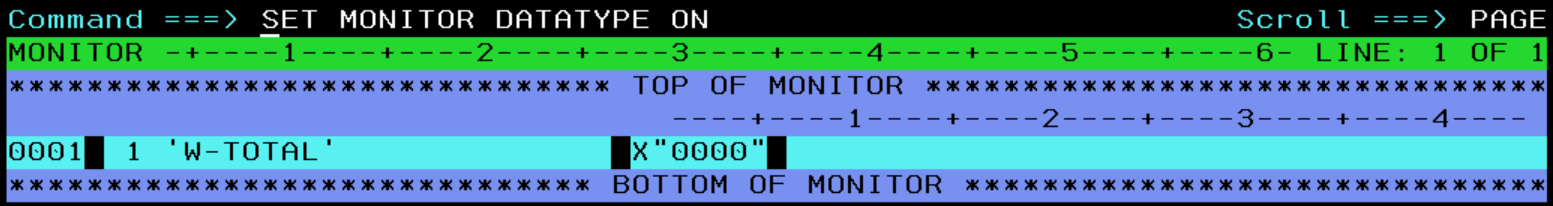
Entre com o comando SET AUTO ON BOTH<Enter>.

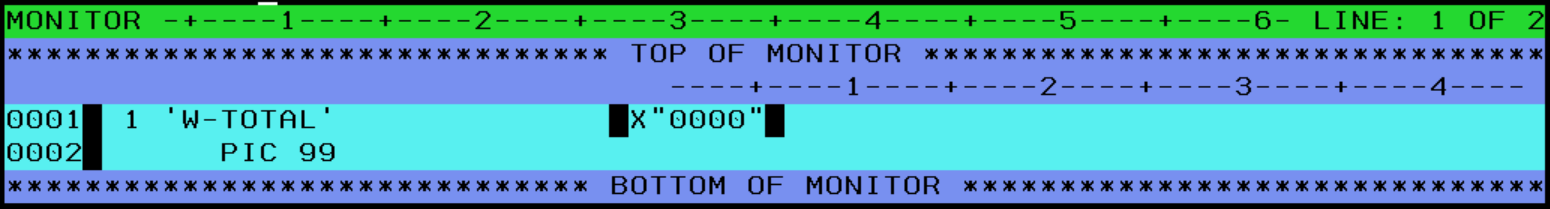
De acordo com as variáveis envolvidas no comando atual e anterior a janela de monitor vai mudando, mostra a variável do comando anterior e o atual antes da execução do comando.





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 |

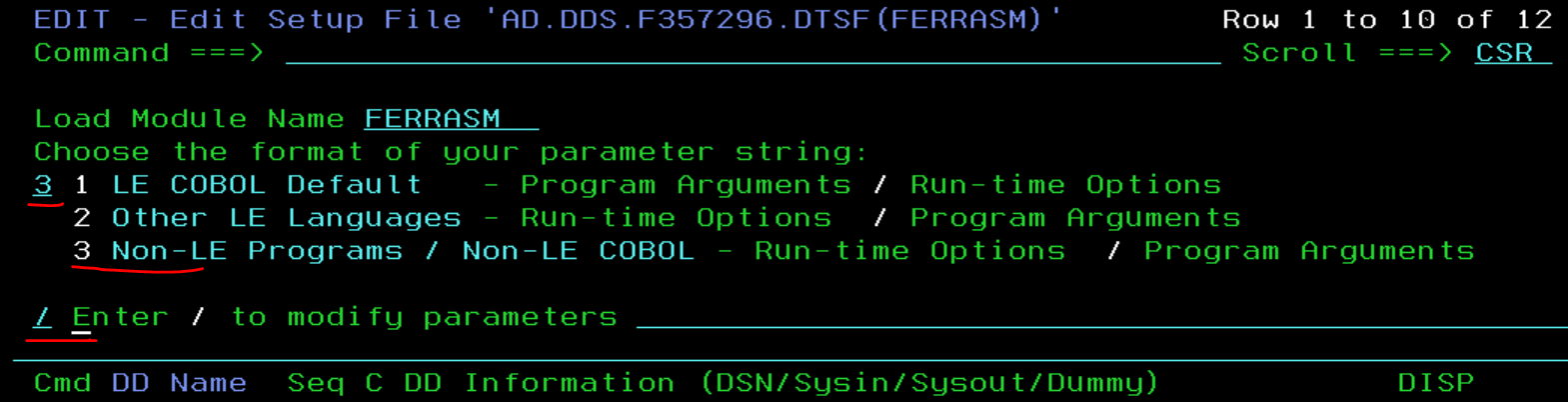
OBs.: para fechar a janela MEMORY aberta com o comando ZOOM MEM, utilize a <Pf10>.

# **Informando parâmetros para a execução – Foreground – Assembler.**

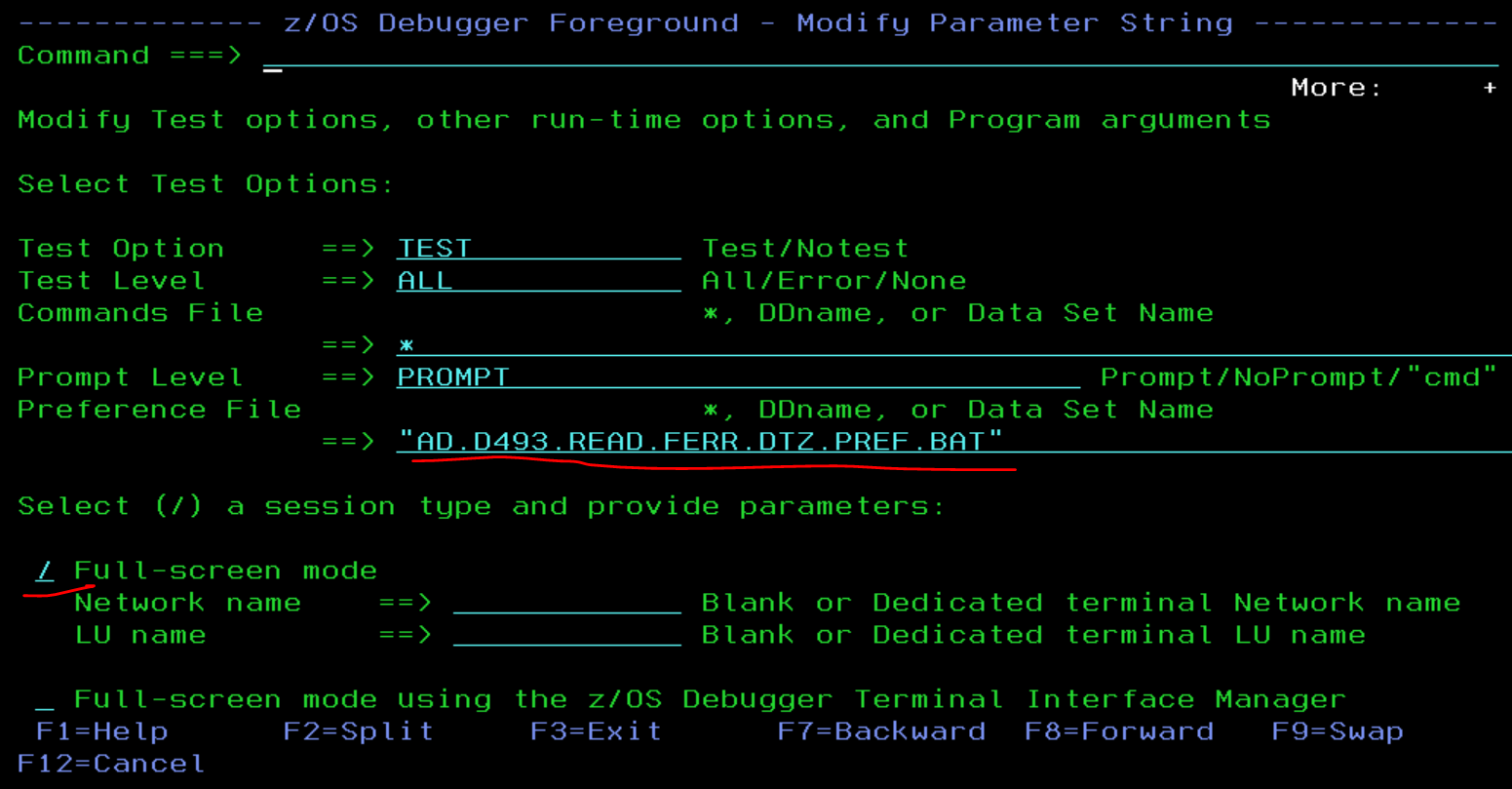
O processo de captura de JCL para execução do batch é o mesmo já descrito em “[Capturando um JCL para execução do programa](#_Capturando_um_JCL)”, após isso, é necessário informar onde encontrar a listagem de compilação do seu programa, essa listagem é gerada no momento da compilação (stage) do Changeman quando marcamos com “/” o campo “Debug Tool”.

Não esqueça de fazer o Promote do programa, pois a versão do seu objeto e da listagem devem estar sincronizados.

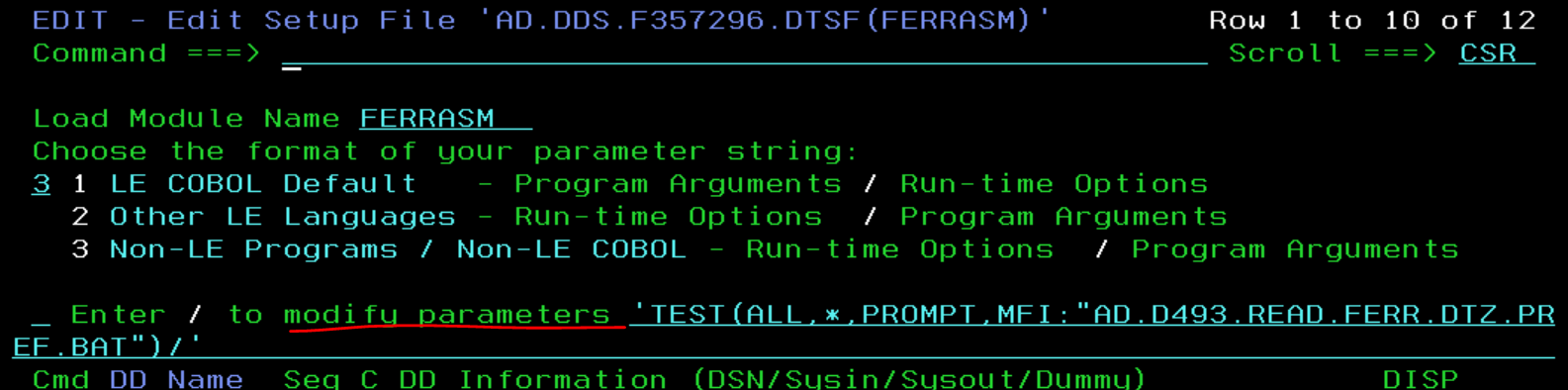
Escolha a opção **3 (“Non-LE Programs / Nom-LE COBOL”)** e selecione com “/” o campo “Enter / to modify parameters”.



Informe o arquivo **"**AD.D493.READ.FERR.DTZ.PREF.BAT**"**, atenção sempre entre **“” (aspas duplas)** e selecione a opção “Full –screen mode” com “/”, digite <Enter> e em seguida <Pf3> para sair.

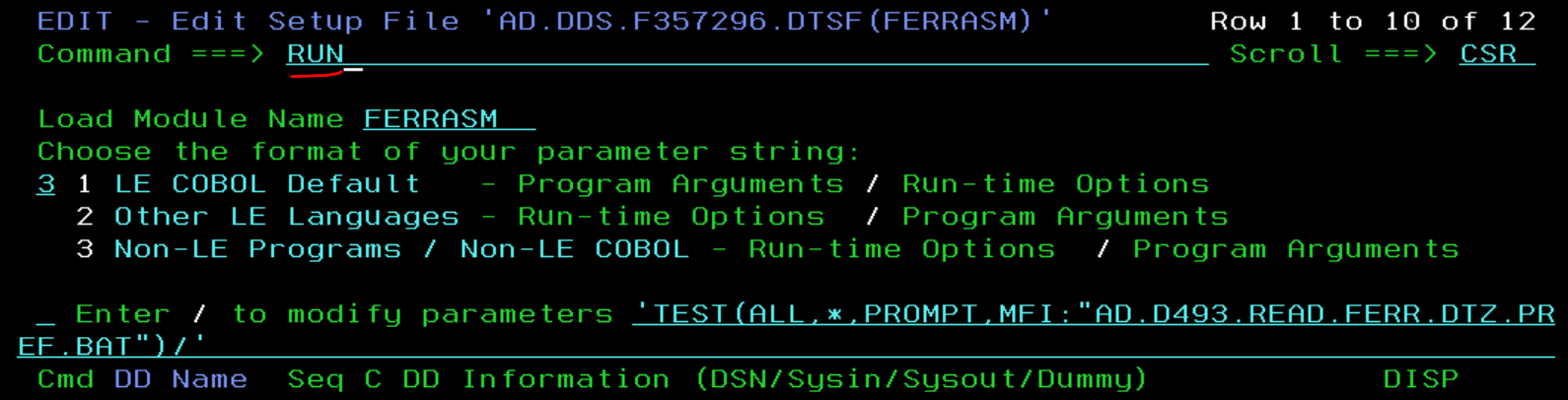


Note que o campo “modify parameters” foi alterado de acordo com a seleção feita.

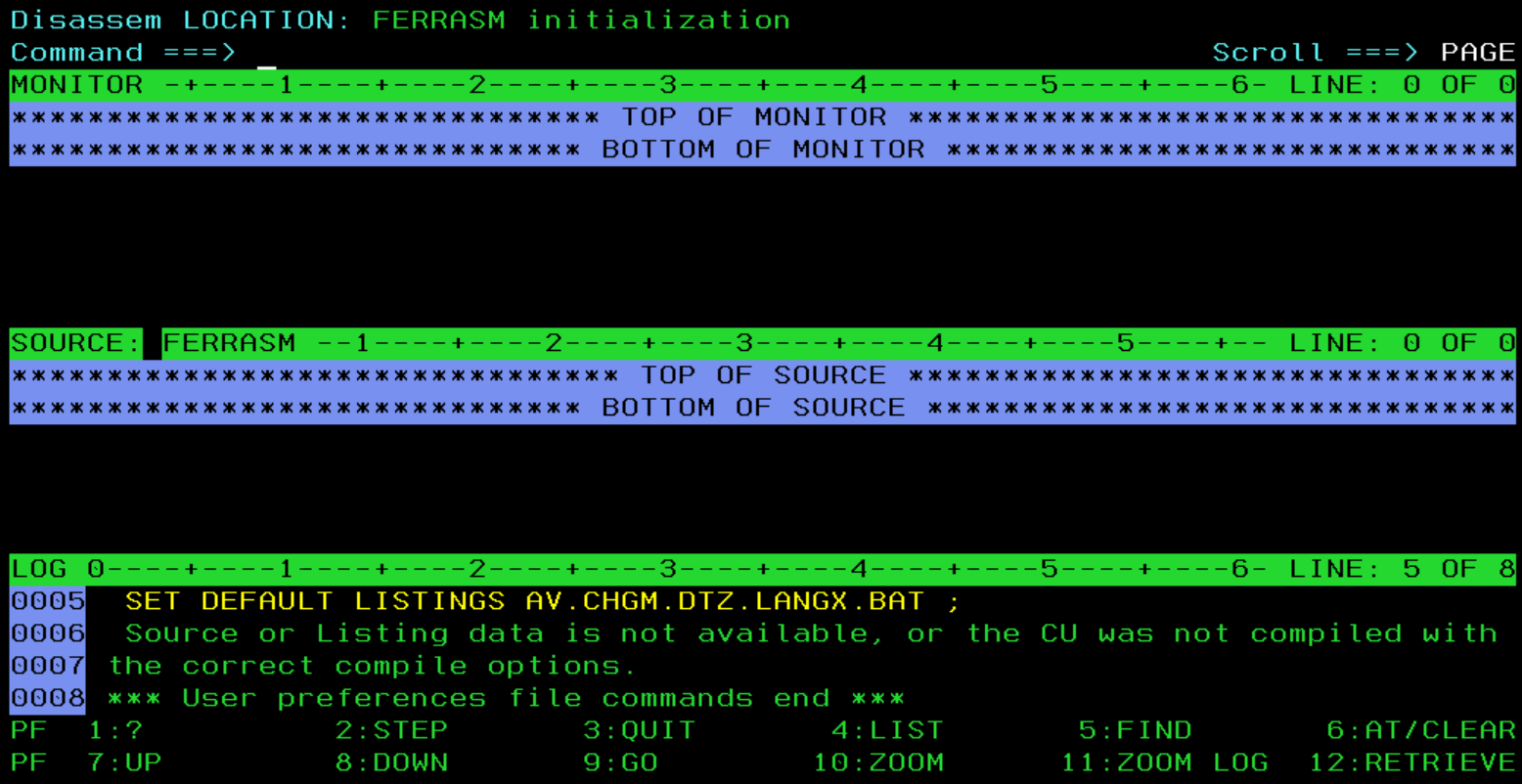


# **Executando um programa com o Debug Tool – Foreground - Assembler.**

Digite “RUN” na linha de comando.

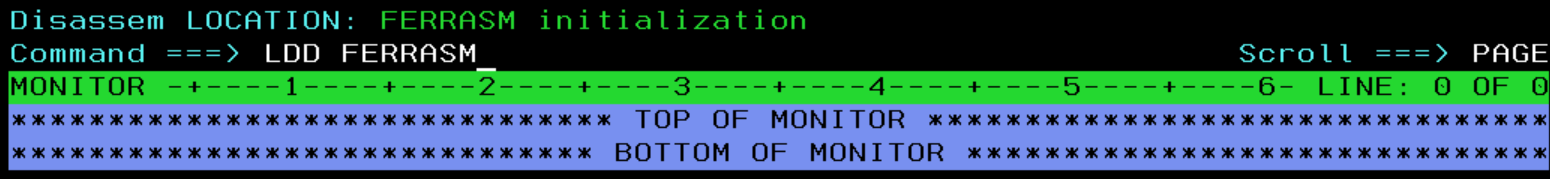


Será exibida a tela seguinte:

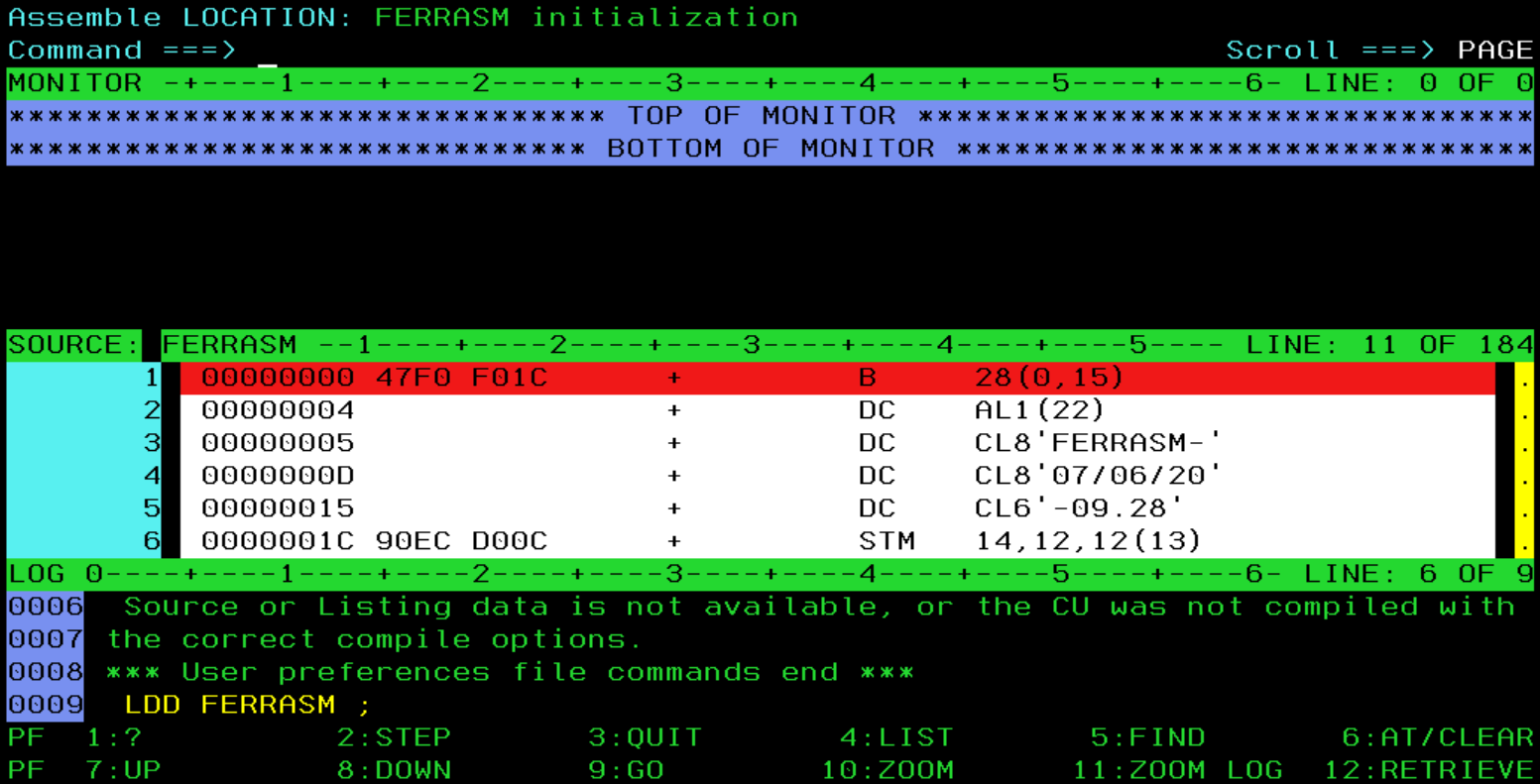


Para instruções sobre as janelas desse painel acesse o item “[Painel padrão do Debug Tool](#_Painel_padrão_do).”.

Faça a carga da listagem do programa através do comando LDD (Load Debug Data).



Fonte do programa exibido na janela Source.



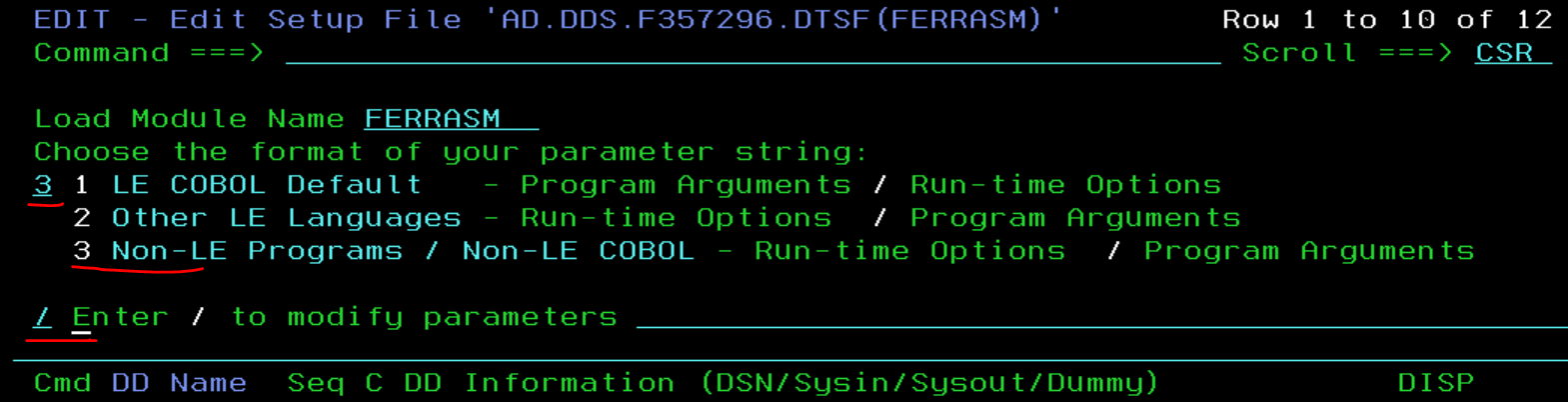
Para prosseguir com o Debug, veja o item “[Descrição das PF´s](#_Descrição_das_PF´s.)”.

# **Informando parâmetros para a execução – Background - Assembler.**

Necessitamos informar ao Debug Tool onde encontrar a listagem de compilação do seu programa, essa listagem é gerada no momento da compilação (stage) do Changeman quando marcamos com “/” o campo “Debug Tool”.

Não esqueça de fazer o Promote do programa, pois a versão do seu objeto e da listagem devem estar sincronizados. (conforme descrito no item [Preparando programa para Debug no Changeman](#_Preparando_programa_para).)

Escolha a opção **3 (“Non-LE Programs / Nom-LE COBOL”)** e selecione com “/” o campo “Enter / to modify parameters”.

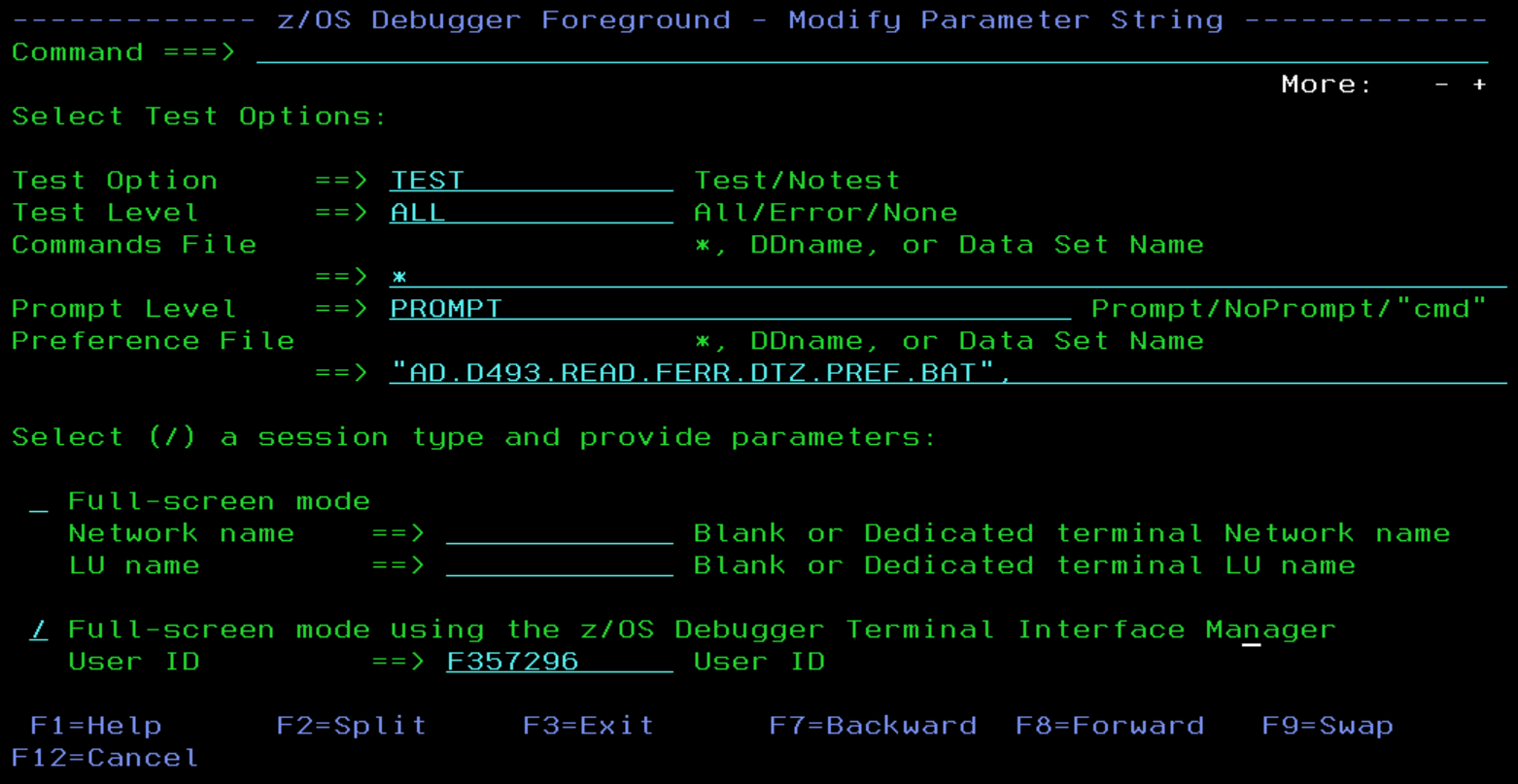


Note que o campo “modify parameters” foi alterado de acordo com a seleção feita.



Neste caso o JOB de execução será submetido por essa sessão TSO e o Debug será feito através do TIM (Terminal Inteface Manager) associado ao “User ID” informado.

Informe o arquivo **"**AD.D493.READ.FERR.DTZ.PREF.BAT**"**, atenção sempre entre **“” (aspas duplas)** e selecione a opção “Full-screen mode using the z/OS Debugger Terminal Interface Manage” com “/”, digite <Enter> e em seguida <Pf3> para sair.



# **Executando um programa com o Debug Tool – Background - Assembler.**

Na sessão TSO fazer a execução do programa com o comando RUN (ou <Pf4>) que mantém a sessão TSO presa enquanto durar o Debug, ou através do comando SUBMIT (OU <Pf10>) que libera o TSO.



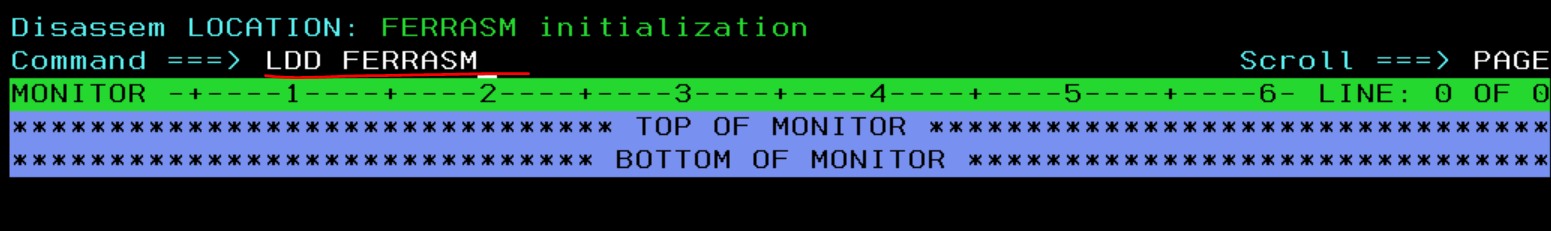
Voltar para a sessão onde o TIM está aberto, deverá estar na tela do Debug.

Caso tenha feita a opção pelo comando SUBMIT (OU <Pf10>), será aberta a tela abaixo com o JCL para execução, que deve ser comandado através do SUBMIT.



Para acompanhar a execução volte para a sessão onde está aberto o TIM.

Será necessário fazer a carga da listagem do programa através do comando LDD (Load Debug Data).

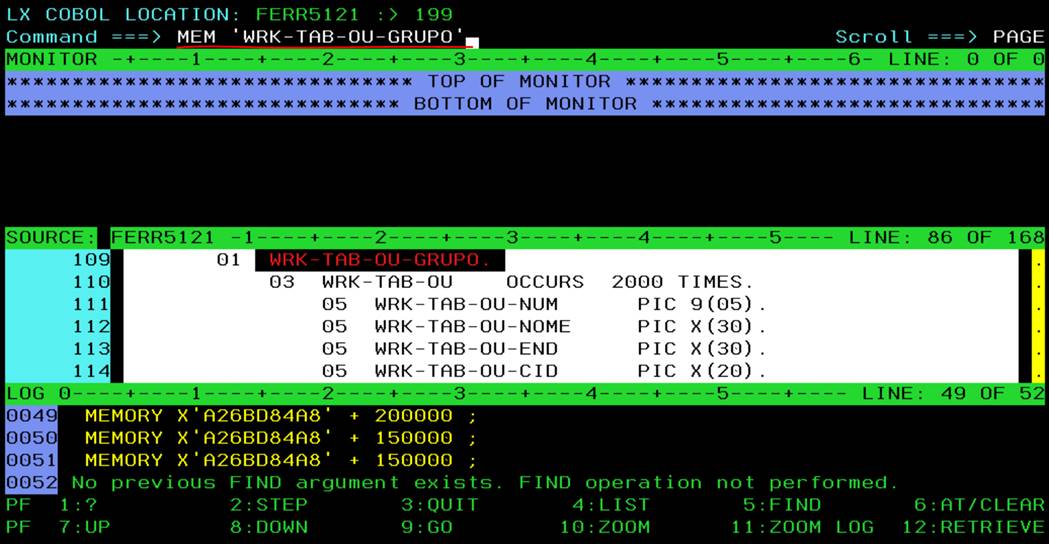


Dar continuidade no Debug utilizando os comandos e PFs descrito no item [Descrição ds PF´s](#_Descrição_das_PF´s.).

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

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

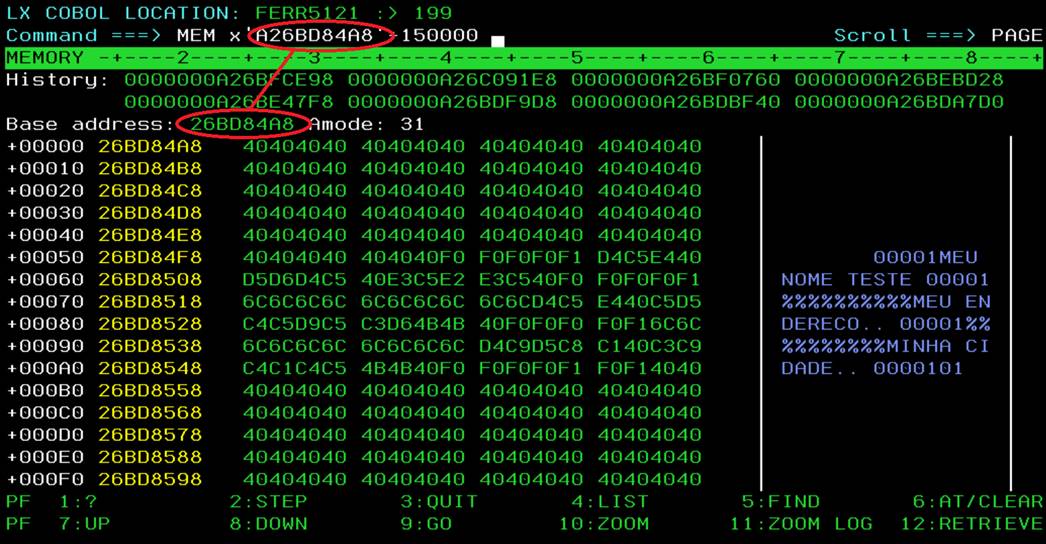
**Comando: MEM 'WRK-TAB-OU-GRUPO'**

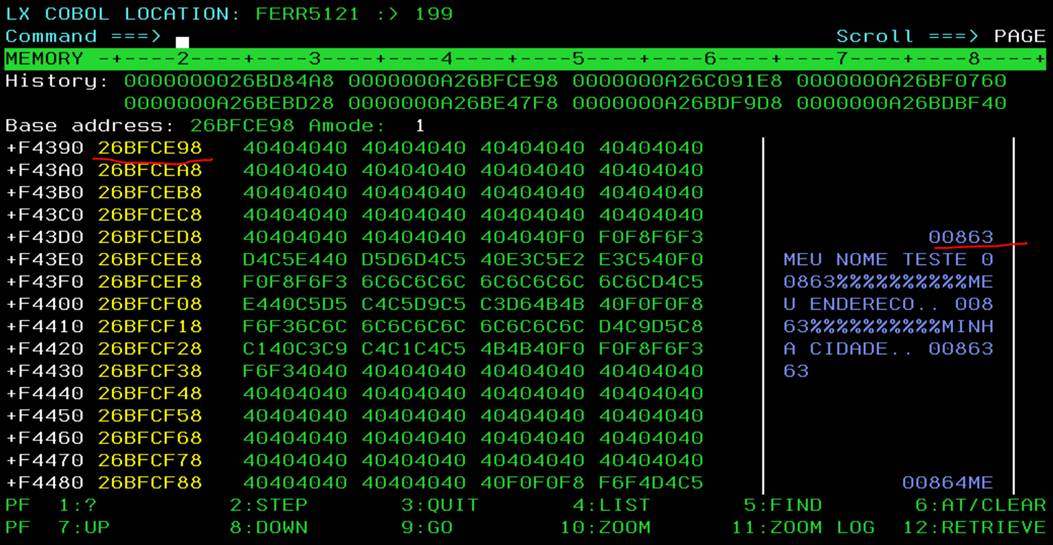


**Comando: ZOOM MEM (para abrir a janela MEMORY)**



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



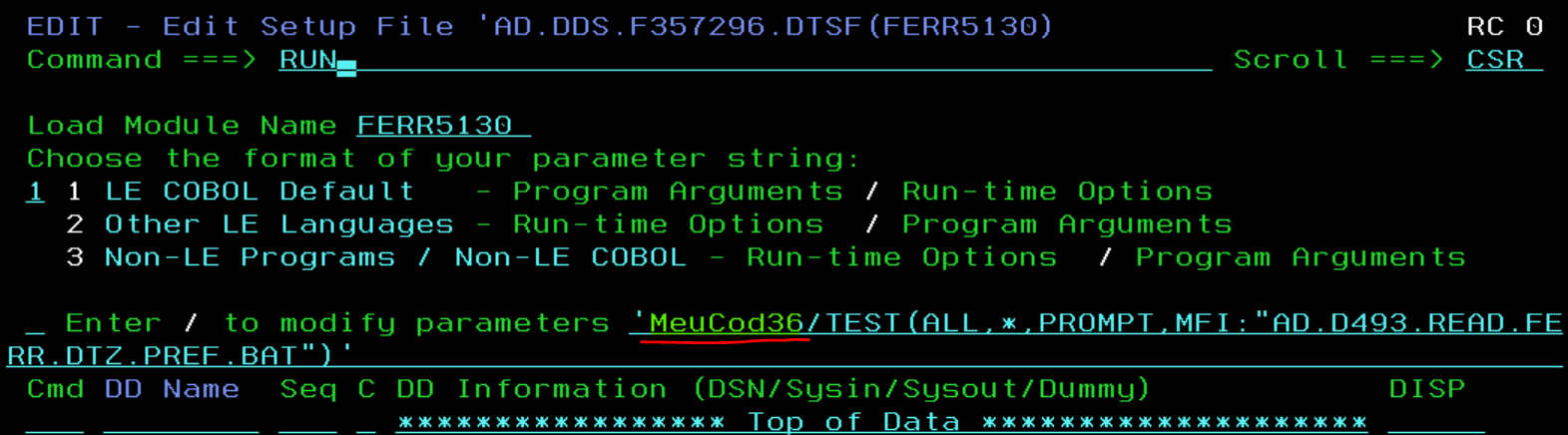


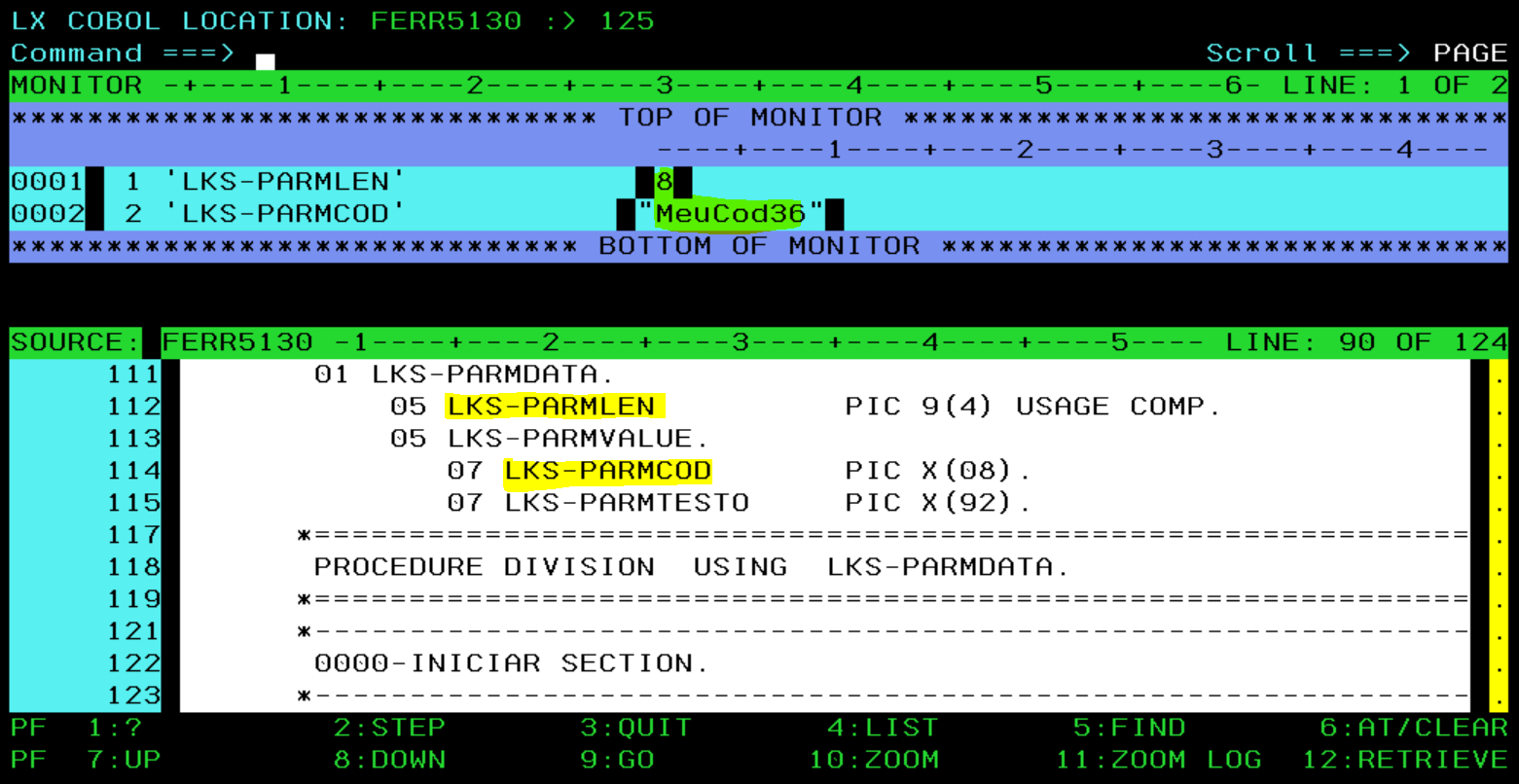
**<PF10> (ZOOM) para fechar a janela MEMORY.**

# **Apêndice II – Passando parâmetros via PARM.**

Para passar um parâmetro para um programa via PARM, informe o parâmetro antes da barra “/” do parâmetro “/TEST” no campo “modify parametres”.

Obs.: o tamanho máximo para passagem de parametro via PARM é de 100 bytes.





# **Apêndice III – 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:

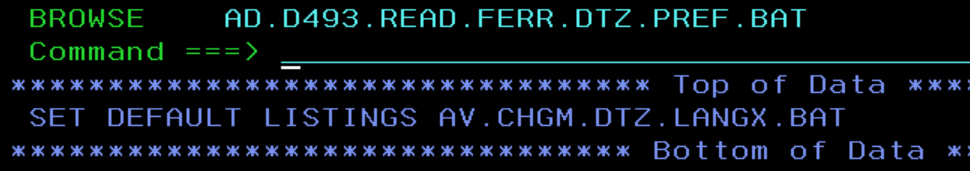
Organization . . . : PS

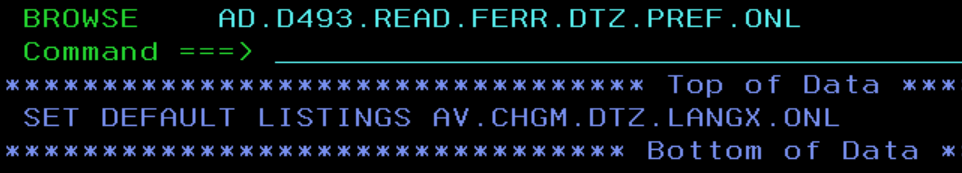
Record format . . . : FB

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>