

# Chapter Twenty Two

## THE SURPAS WORKBENCH - PART 3

### Introduction

This chapter describes additional capabilities and some of the tools available in the SuRPAS Workbench application. This chapter is a completion of the discussion which began in chapters seven and nineteen.

### Objectives

To take advantage of the capabilities of the SuRPAS Workbench application, a developer should be able to:

- Design code that implements and manages parameter settings
- Understand and use the CMS code management utility
- Use the cross reference capability
- Become familiar with the file conversion utilities
- Better understand how to examine database file data

## Navigating the Workbench Main Menu

The main menu displayed when the Workbench application is entered allows you to enter a three character selection that matches the functions and tools supported. The main menu is displayed below. Notice that the top line, shown in reverse video, displays your active database. The bottom line indicates that pressing the **PF2** key ends the Workbench session. The KEA! emulator maps the PF keys to the top keys of the numeric keypad - in this case the forward-slash key.

```
***** FAL PROGRAMMERS WORKBENCH -- MAIN MENU          DB : 0
ENTER SELECTION : CMS

CMS - Code Management System
TKN - TKN Definition/Maintenance Menu
JOB - Job Definition/Maintenance Menu
MFS - MFS Definition/Maintenance Menu
PAR - PARAMS Definition/Maintenance Menu
ERX - SuRPAS ERRXLATE/Errors Table
REP - Report Definition/Maintenance Menu
ICS - Interface Definition/Maintenance Menu
SCR - Screen Definition/Maintenance Menu
FIL - FILE Definition/Maintenance Menu
REF - Cross Referencing Menu

A2E - ASCII to EBCDIC File Translation
E2A - EBCDIC to ASCII File Translation
DBM - Database Management
EXM - Examine File
REL - Funds Associates Release System
END - End Session

PF2 - End Session
```

Figure 22-1 - Workbench Main Menu

### Main Menu Selections

The main menu supports seventeen different selections. A number of these selections provide access to other applications, covered later in the training course. Still other selections are no longer used and their original purpose is no longer known.

## The Code Management System

The CMS option in SuRPAS Workbench provides access to the SuRPAS Code Management System. CMS provides access to file delivery and the software trees. SuRPAS supports 10 trees (blank through 9) with 5 slots per tree. The five slots include the three visible slots (Production, Test, and Development) and two unseen client delivery trees (A client and B client).

Selection of the CMS option results in the display of the CMS release system screen seen in Figure 22-2 below. The screen is made up of three components: the software trees at the top of the screen, the software retrofit order in the middle, and the selection menu at the bottom. The menu supports eight options:

1. Check OUT a module
2. Check IN a module
3. Remove status of a module
4. Move software
5. Accept an auto-merge
6. Compile a module
- ? For Help

**Return** to exit the screen

PFPC Global Fund Services Release System									
13.1R	N/A	12.20	12.2R	12.2a	12.1R	11.20	11.2R	N/A	11.2A
P	-	P2	P3	P4	P5	P6	P7	-	P9
T	-	T2	T3	T4	T5	T6	T7	-	T9
D	-	D2	D3	D4	D5	D6	D7	-	-
- SW Retrofits -									
P7 > T7/D7/T6/D6/T5/D5/T4/D4/T3/D3/T2/D2/D							T2 > D2/D		
P6 > T6/D6/T5/D5/T4/D4/T3/D3/T2/D2/D							D7 > D6/D5/D4/D3/D2/D		
P5 > T5/D5/T4/D4/T3/D3/T2/D2/D							D6 > D5/D4/D3/D2/D		
P4 > T4/D4/T3/D3/T2/D2/D							D5 > D4/D3/D2/D		
T7 > D7/D6/D5/D4/D3/D2/D							D4 > D3/D2/D		
T6 > D6/D5/D4/D3/D2/D							D3 > D2/D		
T5 > D5/D4/D3/D2/D							D2 > D		
T4 > D4/D3/D2/D									
T3 > D3/T2/D2/D									
1) Check OUT a module.					4) Move software.				
2) Check IN a module.					5) Accept an auto-merge.				
3) Remove status of a module.					6) Compile a module.				
ENTER OPTION (? FOR HELP RETURN TO EXIT):									

**Figure 22-2 - CMS Main Screen**

- CMS is available to users based on user permissions as defined in FALLOGIN.COM.
- CMS option 4 moves software from one slot to another slot.
- CMS will do immediate DIFFs as it is moving the software, including older dates over newer dates.

## PAR - PARAMS Definition and Maintenance

The PAR option in SuRPAS Workbench provides access to the parameters database file (FAL\$DATINS:PARAMS.DAT) for a given database. When selected the PARAMS main menu, as seen below, is displayed. The menu provides five possible selections that allow for maintenance of the parameters defined in the PARAMS.DAT file.

PAR is also referred to as PARMaint and can be accessed directly with the DCL command RUN FAL\$TOOL:PARMAINT

```
***** FAL PROGRAMMERS WORKBENCH - PARAMS MENU
```

```
ENTER SELECTION : PAR
```

```
PAR - PARAMS Maintenance Byte-editor
```

```
LIS - List of PARAMS Keys
```

```
DUM - Dump of PARAMS File
```

```
EXT - Extract PARAMS records to file
```

```
INC - Include PARAMS records from file
```

```
MEN - Return to Main Menu
```

```
PF2 - Return to Main Menu
```

The first selection is PAR and allows the user to create, modify, or delete a parameter using the PARAMS byte editor. The byte editor allows access to each character in the parameter field. When the PAR maintenance selection is chosen the user is requested to select an action to be taken. There are five choices, as can be seen on the partial Maintenance screen on the next page:

- A - Add a new parameter
- C - Change the contents of an existing parameter
- D - Delete an existing parameter
- V - View the contents of an existing parameter
- Q - Quit the byte editor and return to the PAR Main Menu

```
***** SURPAS : Maintain Params File
```

```
ENTER ACTION (A,C,D,V,Q) : C
```

Once an action is selected the user will then provide the Key name of an existing parameter or the requested name for a parameter to be created. The name is usually 12 characters in length, but this is not a requirement.

When the **View** action is selected, the screen will then display the contents of the parameter field specified. In the compressed screen displayed below, the STATUSCHECK1 key was selected and a dump of that field is then displayed.

```
***** SURPAS : Maintain Params File
```

```
ENTER ACTION (A,C,D,V,Q) : V
```

```
ENTER KEY           : STATUSCHECK1
```

```
STATUSCHECK1      19991222991999122399N28-JAN-1993 15:24:3912-O  
CT-1993 15:13:3128-JUL-1993 15:33:36
```

The **Add** and **Change** actions require additional information. The user must provide the beginning and ending byte positions of the parameter field being acted on. The contents of the field is then filled or modified.

When adding a new parameter the beginning and ending byte positions define the size of the field to be created. The field is then populated by supplying a set of characters or values in the buffer prompt.

When Changing an existing parameter the beginning and ending byte positions define the specific portion or range of the field to be modified. The buffer prompt displays the existing contents of the range of bytes for the user to modify.

An example of the PARAMS byte maintenance screen can be seen on the next page.

\*\*\*\*\* SURPAS : Maintain Params File

ENTER ACTION (A,C,D,V,Q) : C

ENTER KEY : STATUSCHECK1

ENTER BEG BYTE POSITION : 020

ENTER END BYTE POSITION : 027

ENTER BUFFER : 19991222

PF2 - Cancel

The PARAMS **Extract** selection allows the user to extract parameter records from PARAMS.DAT and write them to another file. This allows the user to then include the parameter records in another PARAMS.DAT file (in another database) using the PAR **Include** selection. An example of the extract support screen is displayed below.

\*\*\*\*\* EXTRACTION OF PARAMS RECORDS

INPUT FILE NAME : PARAMS.KEY

OUTPUT FILE NAME : PARAMS.EXT

## REF - Cross Referencing

The REF option in SuRPAS Workbench provides research and referencing tools for finding: program code, subroutines, command procedures, code and data modules, and files. The REF option provides eight selection options, as shown in the Cross Referencing Main Menu, below. The selections provide the following capabilities:

- **TRE** - Routine Tree - displays the full routine calling tree for a given code module. The display can range from high level (only one level deep of called routines) to a complete tree with all called modules listed to the deepest module.
- **TRS** - Routine Tree Search - allows the user to select a specific code module tree and search for a string within the tree. All instances of the searched string are displayed (or optionally written to a file) with the appropriate location references.
- **LIS** - Routine Listings - lists either all routines that call a user-specified function or subroutine, or all routines that a user-specified function or subroutine calls.
- **LNK** - Listing of Routines that need Linking - this selection lists either the programs that include a specific module, or all modules used by a program or executable image. This selection is particularly useful when you need to know what needs to be re-linked when a program is changed.
- **INC** - Include File Listings - provides a list either of all routines that contain a user-specified include file, or all include files used by a user-specified routine.
- **CER** - Command Files that Execute Routines - lists either command procedures used by a specified program, or programs that are executed within a specified command procedure.
- **REC** - Routines that Execute Command Files - lists either routines that execute a specified command procedure, or command procedures that are executed within a specified routine.
- **CFD** - CFD File Listings - lists the contents of a user-specified .CFD file. This is useful because most .CFD files are stored in text libraries. As such, it is sometimes difficult to find the .CFD file and/or be able to view it (without first extracting it).



The REF Cross Reference Main menu is displayed below.

```
***** FAL PROGRAMMERS WORKBENCH – CROSS REFERENCING MENU

ENTER SELECTION : TRE

TRE - Routine Tree

TRS - Routine Tree Search

LIS - Routine Listings

LNK - Listings of Routines that need Linking

INC - Include File Listings

CER - Command Files that Execute Routines

REC - Routines That Execute Command Files

CFD - CFD File Listings

END - Return to Main Menu

PF2 - Return to Main Menu
```

### **TRS - Routine Tree Search**

The TRS Routine Tree Search allows the user to enter a module name and a search string to generate the appropriate tree. The TRS selection displays screen prompts and an activity log similar to the ones shown below. In this example the user is looking for all instances of “COMMAREA” in the FALINI code module call tree. The results can then be saved in a user directory file with the extension .OUT. A partial set of results are shown on the next page.

```
Module name (eg., FALINI or <CR> to exit): FALINI
Search String: COMMAREA

Generating link map...
Generating search commands...
Search results saved in FALINI.OUT; review (Y/N) [N]
```

```

*****
DISK80:[P4.SOURCE.SUB]FALDEFIOS.FOR;1

        INCLUDE 'FAL$LIBRARY:FILES(COMMAREA.CFD)'
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALDOCMD.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALEXIT.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALFILNAM.* COMMAREA

*****
DISK80:[P4.SOURCE.SUB]FALFILNAM.FOR;1

        INCLUDE 'FAL$LIBRARY:FILES(COMMAREA.CFD)'
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALGETPID.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALINI.* COMMAREA

*****
DISK80:[P4.SOURCE.SUB]FALINI.FOR;1

C                                (3) Clears and initializes
COMMAREA
C                                (4) Sets up COMMAREA fields
        INCLUDE 'FAL$LIBRARY:FILES(COMMAREA.CFD)'
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALIOS.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALOPN.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALRFA.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALSETRAB.* COMMAREA
$  SEARCH /HEAD/NOWARN/EXACT FAL$SUB:FALSRVMOD.* COMMAREA

```

## **LIS - Routine Listings**

The LIS Routine Listings selection provides the user with two listing options, as seen in the Routine Listings screen displayed on the top of the next page.

- LS1 - Provides a list of routines that a routine calls
- LS2 - Provides a list of routines that call a routine
- END - returns the user to the Cross Referencing Main Menu

```
***** FAL PROGRAMMERS WORKBENCH -- ROUTINE LISTINGS
```

```
ENTER SELECTION : LS2
```

```
LS2 - List of Routines that Call a Routine
```

```
LS1 - List of Routines that a Routine Calls
```

```
END - Return to Cross Referencing Menu
```

```
PF2 - Return to Cross Reference Menu
```

When the listing option is chosen a screen will prompt for the routine name and the output option (either display to the screen, write to a file, or both). The display below shows the prompts for the LS2 option.

```
***** FAL PROGRAMMERS WORKBENCH -- LIST OF ROUTINES THAT CALL A ROUTINE
```

```
Please enter the name of the routine (RETURN to exit): FALEXTOPN  
Output option - [S]creen, (F)ile (B)oth:  B
```

A sample of the results that are output by this selection can be found at the top of the next page.

# Routines that use routine FALEXTOPN

-----

ACACNV019	ACCTEDRPT	ACCTRENBE	ACCTRENBR	ACCTVALUE
ACHCOMBINE	ACHSHK	ACSDECODE	ACSUPLOAD	ACTRENBR1
ACTRENBR2	ACTSEARCH	ACTTYP12B1	ADDLPRMS	ADDSRTFLD
ADPACTXRE	ADPACTXRF	ADVANCECOM	ADVCOMCHG	ADVCOMDMP
ADVCOMNEW	AECONFIRM	AERECNSC	AESEN005	AESENNFS
AEWRTNSC	AFF12B1NPD	AFF12B1RPT	AFRACC01	AFRASC01
AFRASR01	AFSCONFIRM	AFSCRETAG	AFSEARREJ	AFSRTIPROC
ALLMENUS	ALTSSNCCR	AMLALTPAY	AMLEXTRPT1	AMLFULLRD
AMLMTLXFR	AMLRPTTEXT2	AMLSHK	AMLSUSPUR	ANWACTVERR
ANWACTVUPL	ANWFCNPRC	ANWXFRRPT	ANWY2KCNV	API1013
API1014	API3082	API3103	API5001	API5005
API5070	API5080	API6536	ARCFALSHK	ARPSHK
ATBJNLPR3	ATBNTFBEG	ATBRESTART	ATBROLMON	ATB_FIX_OUTOFBAL
AUD12B1RPT	AUDARC	AUDARCRPO	AUDFIX	AUTHDEL
AUTHUSERE	AUTHUSP	AUTMENVMS1	AUTMENVMS2	AVGASSCAL
AVGCSTDNL	AVGELGCST	AXAFSAGG	AXAFSAPI	AXDELRPT
AXFNDXRF	AXLOD401	AXPROCESS	AXREDNSC	AXRSHK
AXSENERR	AXSETCOR	AXSETFIL	AXSFBEXT	AXSFBSEN
AXSWEEPS	B12SHK	BALCNVRPT_CNV	BATPURGE	BEBFSEN

Press RETURN to continue or 'X' to exit...

## **REF - Link Listings**

The REF Link Listings selection provides the user with two listing options, as seen in the Link Listings screen displayed below.

- LK1 - Provides a list of programs using a module
- LK2 - Provides a list of modules used by a program
- END - returns the user to the Cross Referencing Main Menu

```
***** FAL PROGRAMMERS WORKBENCH – ROUTINE/PROGRAM LISTINGS
```

```
ENTER SELECTION : LK2
```

```
  LK1 - List of programs using a module
```

```
  LK2 - List of modules used by a program
```

```
  END - Return to Cross Referencing Menu
```

```
PF2 - Return to Cross Reference Menu
```

## A2E and E2A - File Translation Utilities

The SuRPAS Workbench supports two file translation utilities. The A2E ASCII to EBCDIC and the E2A EBCDIC to ASCII file conversion utilities are selected from the Workbench Main Menu. The compressed screens displayed below show the two prompts required by each of the utilities to translate files. In each option the user provides the name of a file to be translated and the output file that will receive the translated version of the input file.

```
***** FAL PROGRAMMERS WORKBENCH - ASCII TO EBCDIC
```

```
ASCII INPUT FILE NAME      :
```

```
EBCDIC OUTPUT FILE NAME   :
```

```
***** FAL PROGRAMMERS WORKBENCH - EBCDIC TO ASCII
```

```
EBCDIC INPUT FILE NAME    :
```

```
ASCII OUTPUT FILE NAME    :
```

## The Database Management Utility

```
***** FAL PROGRAMMERS WORKBENCH -- DB MANAGEMENT MENU
```

```
ENTER SELECTION : DEL
```

```
DEL - DELETE remove and disable a database
```

```
CPY - COPY load files from existing db
```

```
INT - INITIALIZE start a db with empty files
```

```
CDB - CUSTOMIZE customer database after resto
```

```
VER - VERSION change SW/DB version #
```

```
FIL - FILES existence verification in db
```

```
UPG - UPGRADE perform database upgrade
```

```
PF2 - Return to Main Menu
```

The Database Management Utility supports seven options:

**DEL** - Delete - disables and removes and database (the user provides the database slot #)

**CPY** - Copy - loads files from an existing database into another database slot

- √ all users must be off of both databases
- √ all SuRPAS files will be deleted in the destination database
- √ destination database must be correctly set up and all logicals must exist

**INT** - Initialize - starts a database with empty files

**CDB** - Customize - initializes a customer-specific database after it is restored

**VER** - Version - changes the database or software version #

- √ Select SWC to change the software version #
- √ Select DBC to change the database version #

**FIL** - Files - verifies the existence of particular files in a database

**UPG** - Upgrade - upgrades a database to a later version (one version at a time)

## EXM - Examining Files

```
**** FILE EXAMINE FUNCTION : RECORD PARAMETER INPUT ID: WBEXMKEY
```

```
INPUT FILE NAME      : MASFAL                               LIKE      :
```

```
KEY ID               : 00
```

```
KEY VALUE            : 50
```

```
NUMBER OF RECORDS    : 999999
```

```
PF2 - Cancel
```

- To begin examining data in a database file enter the file name (e.g. MASFAL). For normal SuRPAS database files the file name is all that is needed (e.g, for FAL\$DATMAS:MASFAL.DAT, use MASFAL)
- The LIKE field is used to map a copy of a SuRPAS file to a “unknown” or regular file, thus allowing the use of standard variable names. Place the SuRPAS file name in this field.
- Once the input file name is supplied more information is required: Inputting an indexed file results in the following information prompts:
  - Key ID:** 0 = primary, 1, 2, ... = alternate keys
  - Key Value:** the record whose key is > or = to this value is read first
  - # of Records:** maximum # of records to display

Inputting a sequential file results in the following information prompts:

**Start Record:** the first record # to be read

**# of Records:** the total # of records to read



\*\*\*\*\* FILE EXAMINE FUNCTION : FIELD NAME INPUT

ID : WBEXMNAM

INPUT FILE NAME : MASFAL

LIKE :

KEY ID : 00

KEY VALUE : 50

NUMBER OF RECORDS : 999999

OUTPUT OPTION : S ((S)CREEN, (R)EPORT, (B)OTH)

NAME 01 : MASACT	NAME 09 :	NAME 17 :
NAME 02 : MASNA1	NAME 10 :	NAME 18 :
NAME 03 : MASTYP	NAME 11 :	NAME 19 :
NAME 04 :	NAME 12 :	NAME 20 :
NAME 05 :	NAME 13 :	NAME 21 :
NAME 06 :	NAME 14 :	NAME 22 :
NAME 07 :	NAME 15 :	NAME 23 :
NAME 08 :	NAME 16 :	NAME 24 :

PF1 - Finish;

PF2 - Cancel;

The user is prompted to enter up to 24 variables to be displayed:

- EXM will display an error message if an unknown variable name is entered
- Use the variable names that are declared in the .CFD file

\*\*\*\*\* FILE EXAMINE FUNCTION : SELECTION CRITERIA

ID : WBEXMSEL

INPUT FILE NAME : MASFAL

LIKE :

KEY ID : 00

KEY VALUE : 50

NUMBER OF RECORDS : 999999

OUTPUT OPTION : S ((S)CREEN, (R)EPORT, (B)OTH)

MASTYP EQ '101'

PF1 - Finish;

PF2 - Cancel;

PF3 - Next Screen

Enter a set of criteria to further specify the contents of the data results.

- Example #1: MASTYP EQ '101'
- Example #2: MASACT GE '02' AND LE '06'

\*\*\*\*\* FILE EXAMINE FUNCTION : SELECTION CRITERIA ID : WBEXMSEL

INPUT FILE NAME : MASFAL LIKE :

KEY ID : 00

KEY VALUE : 50

NUMBER OF RECORDS : 999999

MASACT	MASNA1	MASTYP
50582042	JERRY LEWKOWITZ	101
506658145	SCOTT SALTZMAN	101
508456973	IRWIN GREIFF	101
508625302	RITA A KERN	101
5401	Account 1	101
5402	Account 2	101
5404	Account 4	101
5405	Account 5	101
5405001409	MR BILL GOLF	101
5405001508	MR GREEN PACKER	101

PF1 - Continue; PF2 - Quit; PF4 - Catalog; -

Results generated by the EXM utility.