Chapter Nine

THE SURPAS FILE ENVIRONMENT

Introduction

One of the keys to mastering the SuRPAS environment is gaining an understanding of its software and database environment.

This lesson covers the SuRPAS software and database file and directory structure. It also includes an introduction to the LNM utility, which provides logical name access to the database and software.

Objectives

To access a database and its associated software, a user should be familiar with SuRPAS software and databases, and understand the following:

- The LNM Utility and its associated commands.
- The software and database file and directory structure.
- The SuRPAS database internal structure.

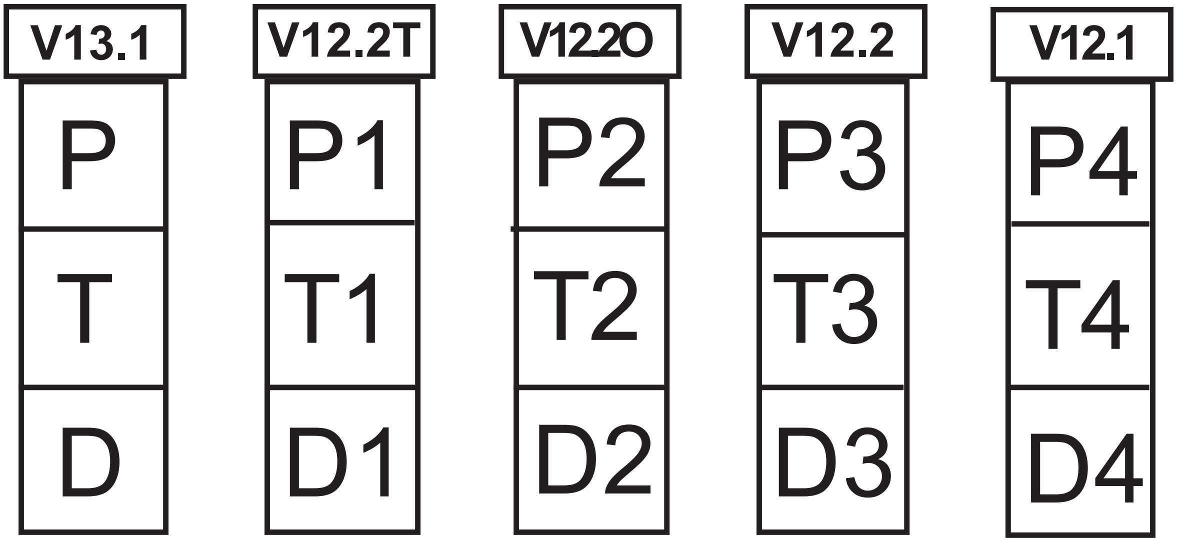
SuRPAS Software

Multiple versions of SuRPAS software are being developed, maintained, and customized at any given time. To support this development, ten generations of SuRPAS software are maintained and accessible by SuRPAS developers and QA testers. Each of these ten generations is supported in a full development, test, and production environment, and each is assigned a number (generation 9 is the least recent environment and the 0 or unnumbered environment is the current one). Using this naming mechanism, the version of SuRPAS that is currently being coded is named *P* for *production*, *T* for *testing,* and *D* for *development*. The next generation is named *P1*, *T1,* and *D1*, and so on.

Version numbers of SuRPAS software contain three components. The whole number portion represents the year (11 = 2001, 12 = 2002, etc.). The fractional portion indicates the release within the year (x.1 is the spring release and x.2 is the fall release). An optional alphabetic character may represent a feature upgrade due to an internal need or requested by a client. It may also indicate whether the version is an *on-release* (*R*), *off-release* (*O*), or *tax release* (*T*) version.

A new version, which can be created at any time, is placed at the appropriate generation level. This change can cause other (earlier) versions to slide back to higher generations (i.e., from P3, T3, & D3 to P4, T4, & D4).

An example of the SuRPAS software generation structure is found in Figure 9-1, below.

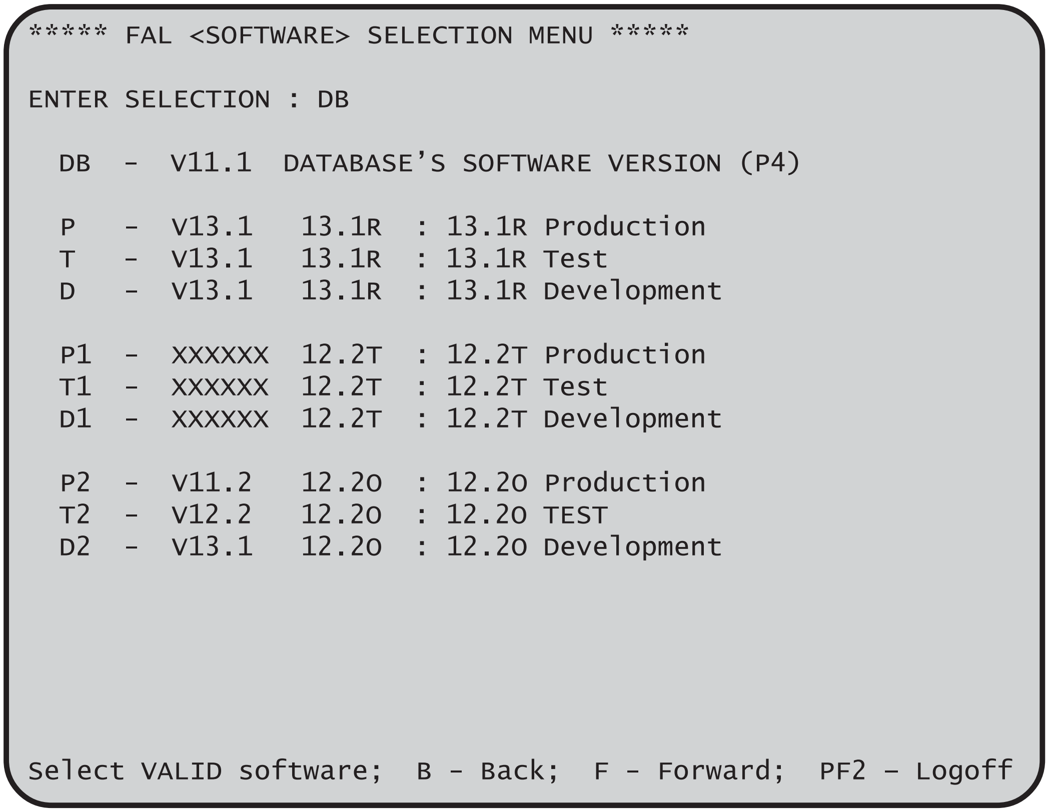


V13.1	V12.2T	V12.2O	V12.2	V12.1
P	P1	P2	P3	P4
T	T1	T2	T3	T4
D	D1	D2	D3	D4

Figure 9-1 - SuRPAS Software Generation Structure

All developers, programmers, and QA testers must declare both the SuRPAS software versions and databases they are currently using. This is accomplished by selecting the appropriate generation, and then indicating whether it is the *development*, *test,* or *production* level. The FAL software selection menus allow for the entry of this information; it provides the default generation level for the database selected (the DB selection).

An example of this selection menu is shown below in Figure 9-2. Notice that the menu shows each generation level and the software version assigned to it.



```
***** FAL <SOFTWARE> SELECTION MENU *****

ENTER SELECTION : DB

  DB  -  V11.1  DATABASE'S SOFTWARE VERSION (P4)

  P   -  V13.1   13.1R  : 13.1R Production
  T   -  V13.1   13.1R  : 13.1R Test
  D   -  V13.1   13.1R  : 13.1R Development

  P1  -  XXXXXX  12.2T  : 12.2T Production
  T1  -  XXXXXX  12.2T  : 12.2T Test
  D1  -  XXXXXX  12.2T  : 12.2T Development

  P2  -  V11.2   12.2O  : 12.2O Production
  T2  -  V12.2   12.2O  : 12.2O TEST
  D2  -  V13.1   12.2O  : 12.2O Development






Select VALID software;  B - Back;  F - Forward;  PF2 - Logoff
```

Figure 9-2 - Software Selection Menu

SuRPAS Databases

There are currently four database environmentsupported. MF1 and MF2 are maintained by the SuRPAS development group (the Berwyn Manufacturing Department); MF3 is maintained by the GEA Initiative group; and DL1 is maintained by the Delivery Group:

- MF1 is used by developers/programmers.
- MF2 is primarily used by QA Testers and regression testers.
- MF3 has been recently created for use by GEA developers in Westborough (also Lynnfield and Boston).
- DL1 is reserved and used by the Delivery Group for handling post release testing, performance issues and testbed activities.

The GEA initiative may require the creation of additional environments. The four current environments are shown in Figure 9-3 on the next page.

Each database environment contains 99 slots. These slots are assigned for the use of programmers and QA testers. A slot can contain any version of the SuRPAS database and can be modified for testing purposes. Databases in these slots are not backed up by the regular SuRPAS backup activity. The owner of the slot is responsible for backing up his or her database.

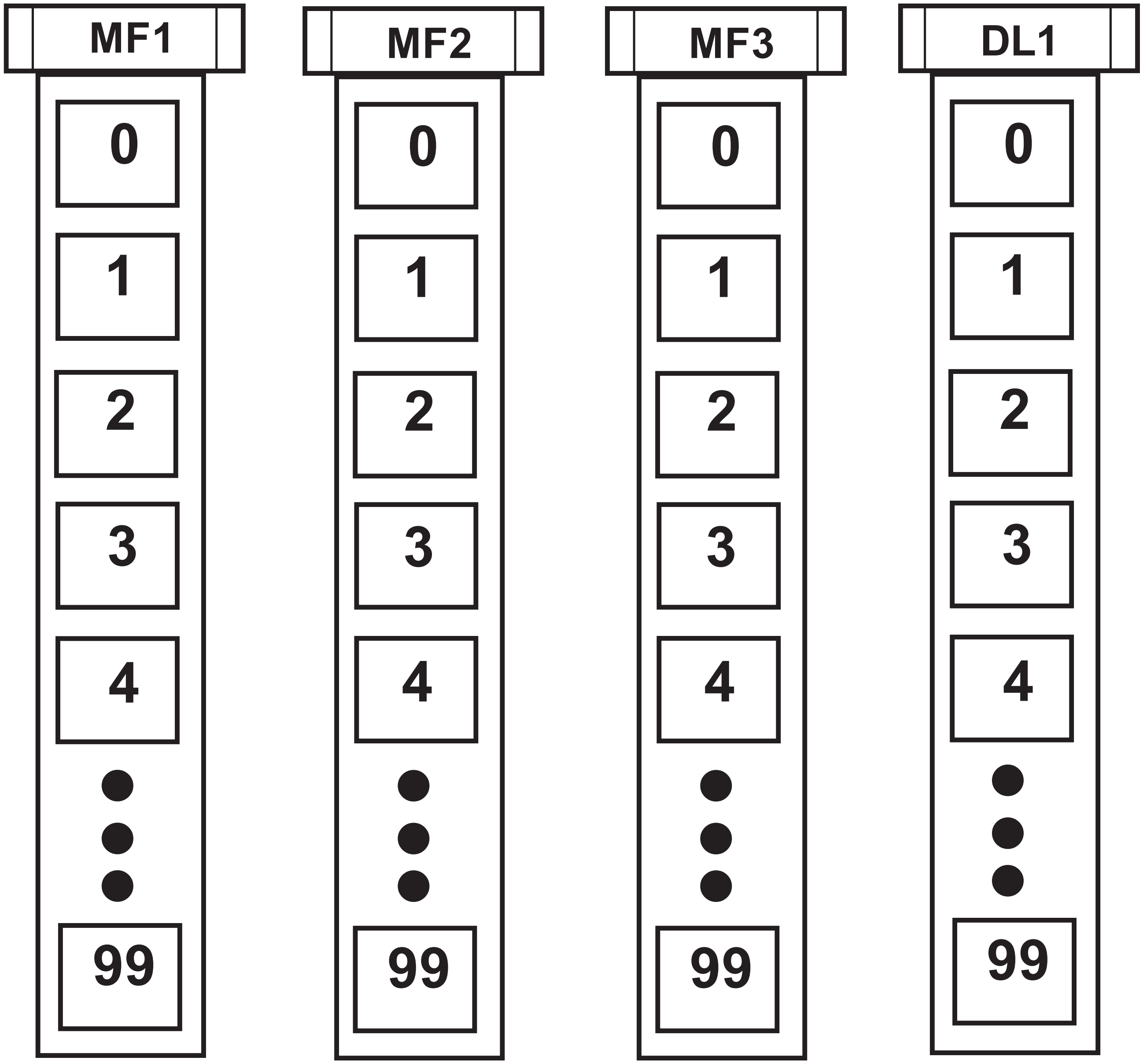
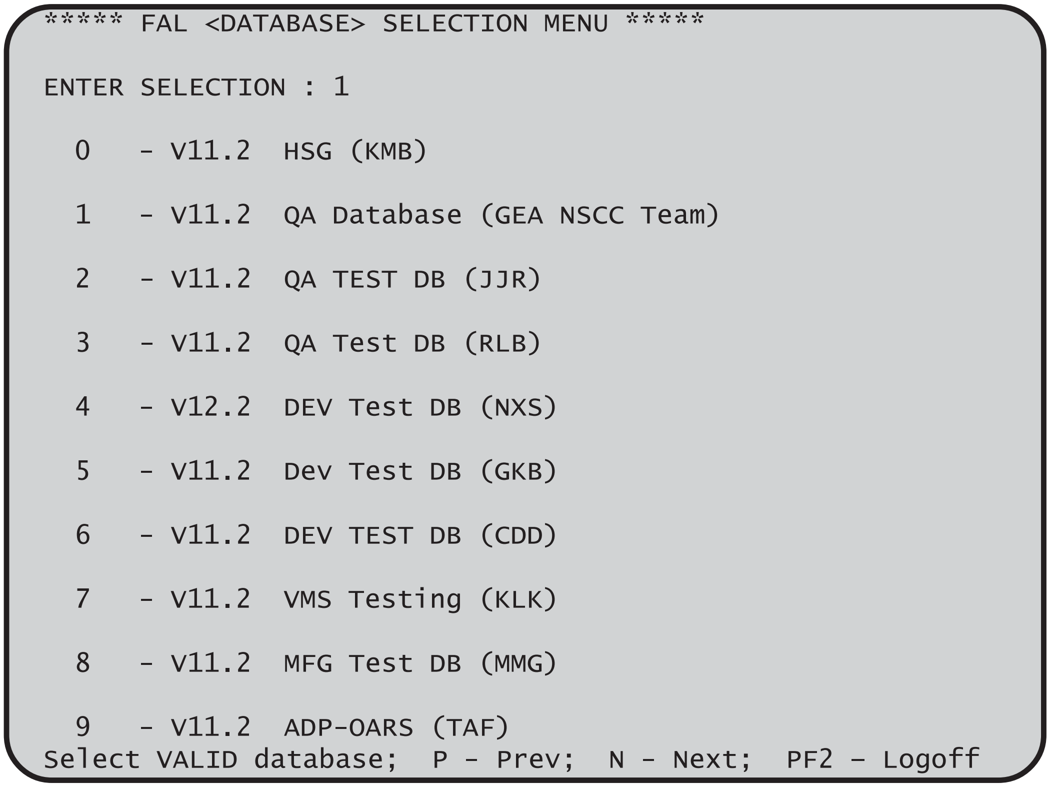


Figure 9-3 - SuRPAS Database Map

When a user logs in to the OpenVMS system, both the database being used and the compatible software version level are selected. Figure 9-4 is an example of the menu that allows the user to select a database slot . Notice that for each database slot number, the menu also shows the following:

- the version of the software that supports the database in that slot,
- what the database is being used for, and
- the TLA (three letter acronym) of the owner assigned to that slot.



```
***** FAL <DATABASE> SELECTION MENU *****

ENTER SELECTION : 1

  0   - V11.2  HSG (KMB)

  1   - V11.2  QA Database (GEA NSCC Team)

  2   - V11.2  QA TEST DB (JJR)

  3   - V11.2  QA Test DB (RLB)

  4   - V12.2  DEV Test DB (NXS)

  5   - V11.2  Dev Test DB (GKB)

  6   - V11.2  DEV TEST DB (CDD)

  7   - V11.2  VMS Testing (KLK)

  8   - V11.2  MFG Test DB (MMG)

  9   - V11.2  ADP-OARS (TAF)
Select VALID database;  P - Prev;  N - Next;  PF2 - Logoff
```

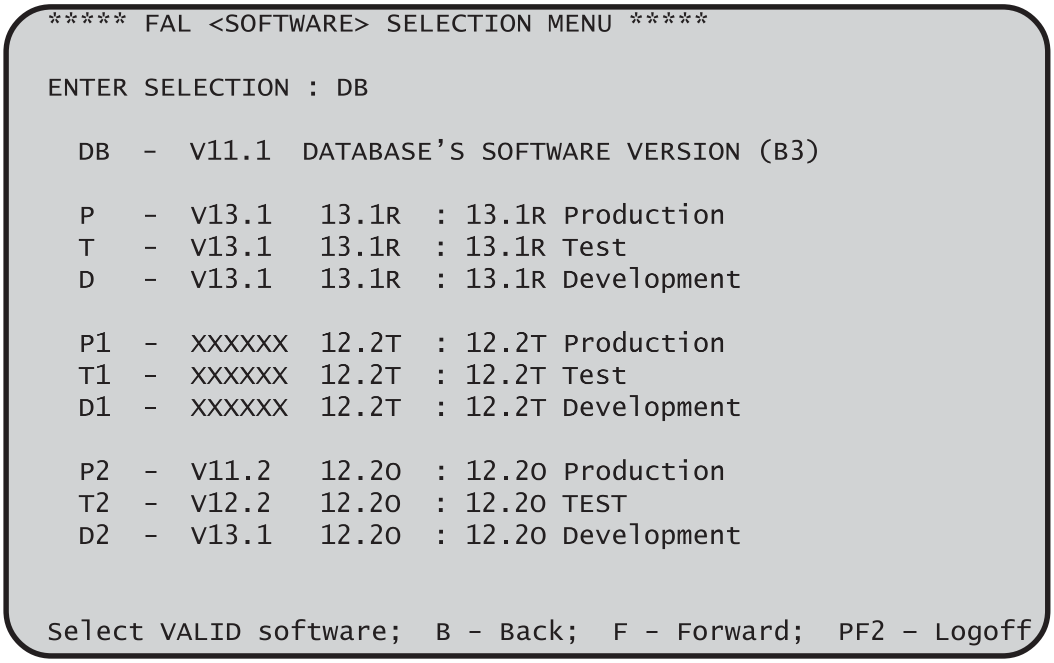
Figure 9-4 - Database Selection Menu

The LNM Utility

The LNM Utility supports the logical name translation activity which manages user access to SuRPAS software versions and database slots within a database environment. When an LNM command is issued, either in a user's LOGIN.COM or as a command at the DCL prompt, a full set of logical name declarations is issued on behalf of the user. These logical name declarations change pointers to both the SuRPAS software and database directories.

The LNM Menus

The LNM MENU command prompts the user for both the database slot number and the software generation. An example of the Database Selection Menu is seen on the previous page (Figure 9-4). An example of the Software Version Selection Menu is shown below in Figure 9-5.

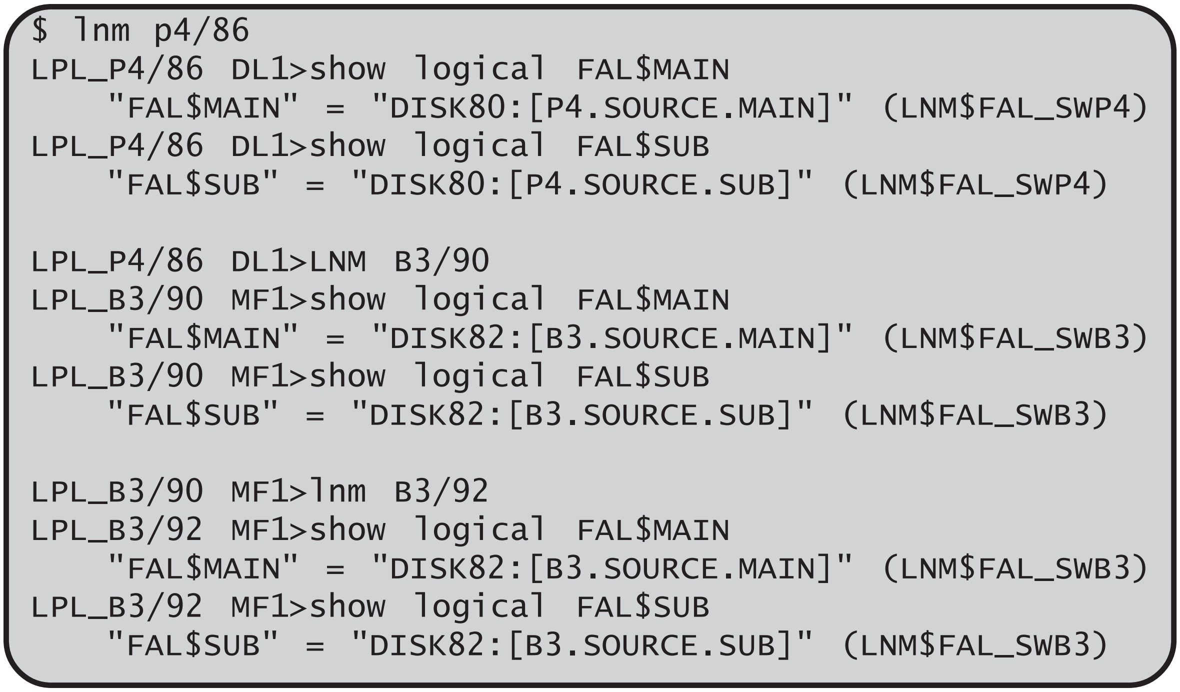


```
***** FAL <SOFTWARE> SELECTION MENU *****

ENTER SELECTION : DB

  DB  -  V11.1  DATABASE'S SOFTWARE VERSION (B3)

  P   -  V13.1   13.1R  : 13.1R Production
  T   -  V13.1   13.1R  : 13.1R Test
  D   -  V13.1   13.1R  : 13.1R Development

  P1  -  XXXXXX  12.2T  : 12.2T Production
  T1  -  XXXXXX  12.2T  : 12.2T Test
  D1  -  XXXXXX  12.2T  : 12.2T Development

  P2  -  V11.2   12.2O  : 12.2O Production
  T2  -  V12.2   12.2O  : 12.2O TEST
  D2  -  V13.1   12.2O  : 12.2O Development


Select VALID software;  B - Back;  F - Forward;  PF2 - Logoff
```

Figure 9-5 - LNM MENU - Software Selection Menu

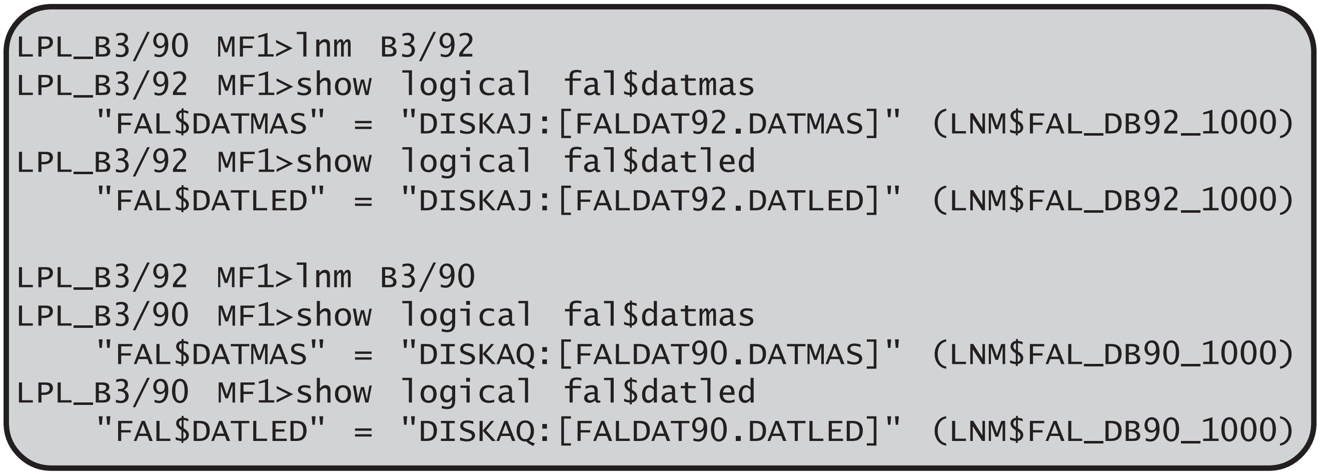
Logical Name Translation

The issuing of any LNM command causes the execution of a series of DCL logical name commands (ASSIGN or DEFINE). The equivalence string for each SuRPAS logical name (see the list on pages 9-10 through 9-13) is changed as a result. The screens below demonstrate these changes.



```
$ lnm p4/86
LPL_P4/86 DL1>show logical FAL$MAIN
    "FAL$MAIN" = "DISK80:[P4.SOURCE.MAIN]" (LNM$FAL_SWP4)
LPL_P4/86 DL1>show logical FAL$SUB
    "FAL$SUB" = "DISK80:[P4.SOURCE.SUB]" (LNM$FAL_SWP4)

LPL_P4/86 DL1>LNM B3/90
LPL_B3/90 MF1>show logical FAL$MAIN
    "FAL$MAIN" = "DISK82:[B3.SOURCE.MAIN]" (LNM$FAL_SWB3)
LPL_B3/90 MF1>show logical FAL$SUB
    "FAL$SUB" = "DISK82:[B3.SOURCE.SUB]" (LNM$FAL_SWB3)

LPL_B3/90 MF1>lnm B3/92
LPL_B3/92 MF1>show logical FAL$MAIN
    "FAL$MAIN" = "DISK82:[B3.SOURCE.MAIN]" (LNM$FAL_SWB3)
LPL_B3/92 MF1>show logical FAL$SUB
    "FAL$SUB" = "DISK82:[B3.SOURCE.SUB]" (LNM$FAL_SWB3)
```

Figure 9-6 - Logical Name Translation of Software Generation Directories



```
LPL_B3/90 MF1>lnm B3/92
LPL_B3/92 MF1>show logical fal$datmas
    "FAL$DATMAS" = "DISKAJ:[FALDAT92.DATMAS]" (LNM$FAL_DB92_1000)
LPL_B3/92 MF1>show logical fal$datled
    "FAL$DATLED" = "DISKAJ:[FALDAT92.DATLED]" (LNM$FAL_DB92_1000)

LPL_B3/92 MF1>lnm B3/90
LPL_B3/90 MF1>show logical fal$datmas
    "FAL$DATMAS" = "DISKAQ:[FALDAT90.DATMAS]" (LNM$FAL_DB90_1000)
LPL_B3/90 MF1>show logical fal$datled
    "FAL$DATLED" = "DISKAQ:[FALDAT90.DATLED]" (LNM$FAL_DB90_1000)
```

Figure 9-7 - Logical Name Translation of Database Directories

The LNM sw/db Command

The single line LNM command changes your software version and database. The syntax is as follows:

```
$ LNM x/#
```

where:

```
x  (software  id)  =  {P,D,T,P1,D1,T1,P2,etc.}
 #  (database  id)   =  {0,1,...,99}
```

This command allows a user to change software and/or database roots without the extra steps required when using the LNM MENU command.

The LNM MENU Command

The LNM MENU command is used to change software and/or database roots via the LNM menus. To enter this command, use the following syntax:

```
$ LNM MENU
```

This will invoke a series of menus similar to those found in Figures 9-4 and 9-5.

Changing Database Environments

To move between environments for a given database, enter one of the following commands:

```
$ LNM _M          (jump to MONTH-END environment)
$ LNM _Y          (jump to YEAR-END environment)
$ LNM _B          (jump to BACKUP environment)
$ LNM _L          (jump to LIVE environment)
```

Software Directories

The following is a list of SuRPAS software directories. The directory specification in a directory structure may include an asterisk. The asterisk is a placeholder for the specific software root (P, T, D, P1, T1, D1, etc.).

1. SuRPAS End-user Software

Directory Logical Name	Description	Directory Structure
FAL$PRGLIB	EXEs & COMs for code SuRPAS	[FALSYS*.PRGLIB]
FAL$TOOL	OBJ & EXE for tools used by SuRPAS staff	[FALSYS*.TOOL]
FAL$STDFIL	“Directory” type files; Reports, Jobs, Tokens, Zip Code	[FALSYS*.STDFIL]
FAL$FILES	FDL (File Definition Language) Repository	[FALSYS*.FILES]
FAL$OBJECT	OBJ for mains and MASTER.OLB for subs	[FALSYS*.OBJECT]
FAL$LIBRARY	Text Libraries (.TLB), WB-REF files	[FALSYS*.LIBRARY]
FAL$PATCH	FTKPKG files	[FALSYS*.PATCH]
FAL$RPTCRD	WB-EXM report layouts	[FALSYS*,RPTCRD]
FAL$SERVER	Server log files & /dsk upgrade packages	[FALSYS*.SERVER]
FAL$WCSDAT	/dsk Work Control System to allow offline trade entry & track who is entering trades from /dsk to ensure all trades get uploaded before Job/process runs. Also contains /dsk Trade Date file.	[FALSYS*.WCSDAT]

NOTE: On an AXP machine the root directory is [FALSYS*_AXP.xxxxxxxx]

2a. SuRPAS Source Code Repository (Programmers)

Directory Logical Name	Description	Directory Structure
FAL$MAIN	"main" (top in tree) programs	[*.SOURCE.MAIN]
FAL$SUB	Subroutines	[*.SOURCE.SUB]
FAL$COM	Command procedures	[*.SOURCE.COM]
FAL$SCR	Screens (.SCR, .FRM, .INC, .FOR)	[,SOURCE.SCREEN]
FAL$MISC	Used by CMS to place "checked-out" modules which are not one of the above (main, sub, com, scr) (e.g. .FIL, .RPT, .JDL, .CFD, .FDL)	[*.SOURCE.MISC]
FAL$TOOLSRC	source code to EXEs & COMs in FAL$TOOL	[*.SOURCE.TOOLSRC]

2b.SuRPAS Source Code Checked in Today Via CMS (Programmers)

Directory Logical Name	Description	Directory Structure
WIP$MAIN	Main routines checked in	[*.WIP.MAIN]
WIP$SUB	Subroutines checked in	[*.WIP.SUB]
WIP$COM	Command procedures checked in	[*.WIP.COM]
WIP$SCR	Screen modules checked in	[*.WIP.SCREEN]
WIP$MISC	Miscellaneous modules checked in	[*.WIP.MISC]
WIP$TOOLSRC	To be used in the future when tools are a part of CMS	[*.WIP.TOOLSRC]
WIP$ARCHIVE	Copy of module from before it was modified (wip$xxxx -> fa;$xxxx -> wip$archive) all iterations of modules kept for given release	[*.WIP.ARCHIVE]

Database Directories

The following is a list of SuRPAS database directories. The directory specification in a directory structure may include a pound sign (#). The pound sign is a placeholder for the database root number (0,1,2,3,4,...,99).

SuRPAS Temporary Data File Software

Directory Logical Name	Description	Directory Structure
FAL$SCRATCH	Work files used during event execution NOTE: JOBPROCESS will delete this entire directory	[FALTMP#.SCRATCH]
FAL$HLDFIL	Primarily used as event backup directory to allow recovery without restoring the entire database NOTE: if you put a file here, you must delete it (e.g., event creates new copy after deleting previous one)	[FALTMP#.HLDFIL]
FAL$REPORT	Reports created by events or UDMS	[FALTMP#.REPORT]
FAL$JE	Log files created during job execution	[FAL.USER.JE#]

SuRPAS Interface Files

Directory Logical Name	Description	Directory Structure
FAL$LOAD	Non NSCC interface files	[FALINT#.LOAD]
SFB$LOAD	NSCC & ANW & AFS interface files	[FALINT#.NSCC]
FAL$EXTRACT	Copies of key database files to be used off-system (ORACLE files)	[FALINT#.EXTRACT]
FAL$DSKEXT	Extract files produced by DSKFILES01 event DSK.MDB (updated hourly) imported by /dsk clients	[FALINT#.DSKEXT]

SuRPAS Site-specific Files

Directory Logical Name	Description	Directory Structure
FAL$FILES_S	Site-specific tuned FDLs created by OPTIMIZE	[FALSIT#.FILES]

SuRPAS Core Database Files (FAL$SEARCH)

Directory Logical Name	Description	Directory Structure
FAL$DATACT	Account routed files (SWP, SPP, SEC, PAY, MEM)	[FALDAT#.DATACT]
FAL$DATARC	Audit trail of clerical account maintenance	[FALDAT#.DATARC]
FAL$DATBAT	Batch & trace & request level files (DAT, DAY, XQF)	[FALDAT#.DATBAT]
FAL$DATEXP	Expense code reporting used by MM funds with drafts (not limited to users of Draft Batch system)	[FALDAT#.DATEXP]
FAL$DATINS	Installation level parameter files (PARAMS, USP, MGT)	[FALDAT#.DATINS]
FAL$DATXCM	External communication parameters	[FALDAT#.DATXCM]
FAL$DATJNL	Posted trades (JNL)	[FALDAT#.DATJNL]
FAL$DATLBL	Labels	[FALDAT#.DATLBL]
FAL$DATLED	Ledgers (LED)	[FALDAT#.DATLED]
FAL$DATMAS	Master (MAS)	[FALDAT#.DATMAS]
FAL$DATMIS	Misc (ACH, PRX, TAX)	[FALDAT#.DATMIS]
FAL$DATNET	NSCC & ANW networking files (DIV, FAC, FCN, BCN)	[FALDAT#.DATNET]
FAL$DATRXT	“Big fat file” (STX) for off-system reporting(Crystal Rpts)	[FALDAT#.DATRXT]
FAL$DATSAL	Sales level files (SAL, DFS)	[FALDAT#.DATSAL]
FAL$DATSCH	Job scheduler files (JBQ, JBX, JPR)	[FALDAT#.DATSCH]
FAL$DATSFB	NSCC Fund/Serv files (REC, ORE, EXC, ORP, SEN)	[FALDAT#.DATSFB]
FAL$DATSIB	Interface processing files (AIF, DAYUPL, PR4, 5, 6)	[FALDAT#.DATSIB]
FAL$DATSUB	Sub-accounting files (DLR, BRN, REP, TIM, AFF, WHL)	[FALDAT#.DATSUB]
FAL$DATTMP	Temporary work files (PR1, 2, 3, 7, CKQ, DSF)	[FALDAT#.DATTMP]
FAL$DATTRN	Transaction files (PND, REJ, LTS)	[FALDAT#.DATTRN]
FAL$DATWIR	Wire order sub-system (WOE, WOP, WOS)	[FALDAT#.DATWIR]
FAL$DATYER	Last year’s transactions (YTD, ACT)	[FALDAT#.DATYER]

Interface File Directories

Interface File Directories are used to hold files that are input and output to and from SuRPAS.

- SFB$LOAD - historically used for NSCC and Mirror interface files. Housekeeping maintenance on the directory is performed by the FILPURDEL event, not the events receiving/creating the files.

- FAL$LOAD - all other interface files. Housekeeping maintenance on the directory is typically performed by the individual events handling the file I/O. Additionally, the FILPURDEL event can be used to maintain the directory.

- FAL$EXTRACT - copies of SuRPAS 'FAL$DATxxx' Standard Data Files that are available for direct access by the client. Again, housekeeping maintenance in this directory is typically performed by the individual extract/copy events, and can be performed by the FILPURDEL event as well.

Temporary File Directories

Temporary File Directories typically remain in existence for the extent of an event or a process period. Note that these directories are NOT handled in the standard BACKUP events.

- FAL$SCRATCH - should be used only for temporary files that are created/ used within the same event. Housekeeping is handled by the JOBSTREAM and DIVPOST events, which delete all files in the directory at the beginning of the event (note that other events may also delete the directory files).

- FAL$HLDFIL - Typically used to store copies of SuRPAS Data Files created at the beginning of events prior to updating the Data Files. These could then be used for both restore and research purposes. Directory housekeeping is not automatic; each event must perform its own file maintenance.

Note that as of October 2000, there is a new once-a-day backup event developed by PSG that will back up FAL$HLDFIL directory.

SuRPAS Standard ‘FAL$DATxxx’ Data File Directories

FAL$DATxxx file directories contain the standard Data Files comprising a SuRPAS database. The directories are included in our standard BACKUP and RESTORE events.

These directories should also be used for temporary files that are created in one event and subsequently used in another event (this usage should be approved by an SA, Annette Swank, or David Vass). Again, these events must perform their own file maintenance. Use discretion with this technique regarding file size and number of versions.

SuRPAS Internal Database Structure

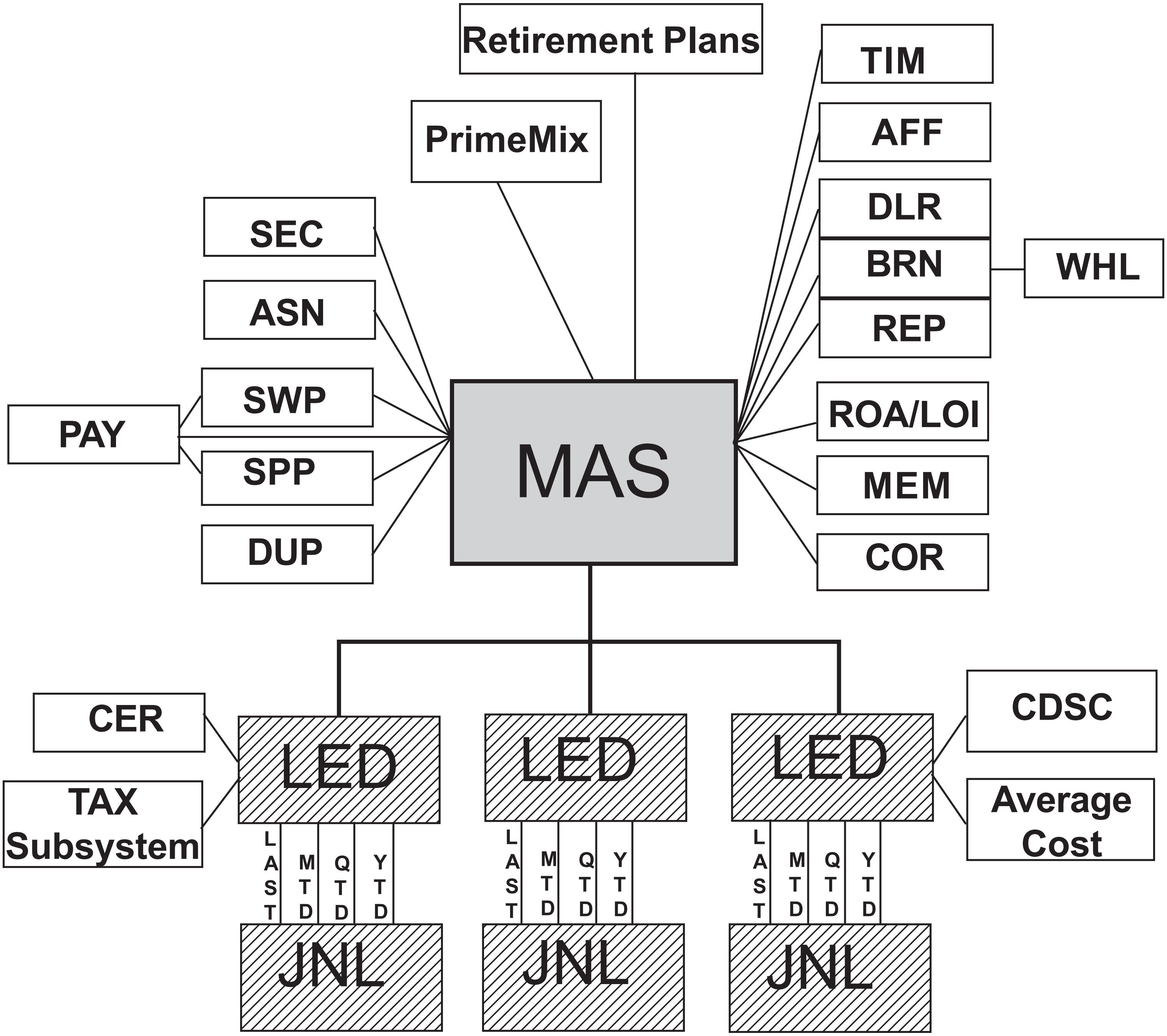


Figure 9-8 - SuRPAS Internal Database Structure

Appendix 9a - SuRPAS Internal Database Structure Definitions

Journal (JNL) - the dollar or share history record of trades in a ledger. SuRPAS stores this year's (JNL), and last year's trades (YTD). Note that the PR3 file is just today's trades, posted in the same format.

Ledger (LED) - A single fund held by a master (client database). SuRPAS allows 20,000 per Master

Master (MAS) - a client that holds multiple fund positions.

PrimeMix - the asset allocation system in SuRPAS.

.